

**SISTEM PENJADWALAN PROYEK  
MENGUNAKAN ALGORITMA GENETIKA**

**TUGAS AKHIR**

Diajukan Sebagai Salah Satu Syarat  
Untuk Memperoleh Gelar Sarjana Teknik Pada  
Jurusan Teknik Informatika

oleh :

**FEBRIA MAHARANI**  
**10851002262**



**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM  
PEKANBARU  
RIAU  
2013**

## **LEMBAR PERSETUJUAN**

### **SISTEM PENJADWALAN PROYEK MENGGUNAKAN ALGORITMA GENETIKA**

#### **TUGAS AKHIR**

oleh:

**FEBRIA MAHARANI**  
**10851002262**


Telah diperiksa dan disetujui sebagai laporan tugas akhir  
di Pekanbaru, pada tanggal 7 Oktober 2013

Koordinator Tugas Akhir



**M. Affandes, ST, MT**  
**NIK. 130 510 030**

Pembimbing



**DR. Okfalisa, ST, M.Sc**  
**NIP. 19771028 200312 2 004**

## LEMBAR PENGESAHAN

### SISTEM PENJADWALAN PROYEK MENGGUNAKAN ALGORITMA GENETIKA

#### TUGAS AKHIR

oleh:

**FEBRIA MAHARANI**  
**10851002262**

Telah dipertahankan di depan sidang dewan penguji  
Sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik  
Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau  
Di Pekanbaru, pada tanggal 7 Oktober 2013

Pekanbaru, 7 Oktober 2013  
Mengesahkan,



Dekan

**Dra. Hj. Yenita Morena, M.Si**  
**NIP. 19601125 198503 2 002**

Ketua Jurusan

**Elin Haerani, ST, M.Kom**  
**NIP. 19810523 200710 2 003**

#### DEWAN PENGUJI

Ketua : DR. Okfalisa, ST, M.Sc  
Sekretaris : DR. Okfalisa, ST, M.Sc  
Penguji I : Elin Haerani, ST, M.Kom  
Penguji II : Elvia Budianita, ST, M.Cs

# **SISTEM PENJADWALAN PROYEK MENGUNAKAN ALGORITMA GENETIKA**

**FEBRIA MAHARANI**  
**10851002262**

Tanggal Sidang: 7 Oktober 2013

Periode Wisuda : 28 November 2013

Jurusan Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Sultan Syarif Kasim Riau

## **ABSTRAK**

Tahap perencanaan dan penjadwalan merupakan tahapan yang paling menentukan keberhasilan suatu proyek. Hal ini dikarenakan penjadwalan adalah tahap ketergantungan antar aktivitas yang membangun proyek secara keseluruhan. Pemecahan masalah penjadwalan yang baik dari suatu proyek merupakan salah satu faktor keberhasilan dalam pelaksanaan proyek untuk selesai tepat pada waktunya. Tugas akhir ini menerapkan algoritma genetika untuk memecahkan masalah optimasi dalam penjadwalan proyek. Algoritma genetika merepresentasikan kandidat solusi penjadwalan kedalam kromosom-kromosom secara acak, lalu dievaluasi menggunakan fungsi *fitness* dan seterusnya dilakukan seleksi. Metode seleksi yang digunakan dalam penelitian ini adalah metode seleksi roda *roulette*, kemudian dilakukan pindah silang dan mutasi. Pada setiap generasi, kromosom dievaluasi berdasarkan nilai fungsi *fitness*. Setelah beberapa generasi maka algoritma genetika akan menghasilkan kromosom terbaik, yang merupakan solusi optimal. Implementasi sistem dalam penelitian ini menggunakan *Hypertext Preprocessing* (php) dan mysql. Pengujian yang dilakukan meliputi pengujian *blackbox*, performansi, dan *user acceptance test*. Hasil dari sistem penjadwalan proyek menggunakan algoritma genetika adalah jadwal kegiatan-kegiatan dalam sebuah proyek yang dapat menjadi alternatif keputusan bagi kontraktor dalam pelaksanaan proyek. Dan berdasarkan hasil pengujian performansi yang dilakukan sebanyak 10 kali dapat disimpulkan bahwa algoritma genetika membutuhkan waktu yang lama jika nilai iterasinya besar, karena dalam algoritma ini terdapat proses penggenerasian.

**Kata kunci:** Algoritma Genetika, Fungsi *Fitness*, Optimasi, Penjadwalan Proyek, Pindah Silang, Mutasi.

# ***PROJECT SCHEDULING SYSTEM USING GENETIC ALGORITHM***

**FEBRIA MAHARANI**  
**10851002262**

*Final Exam Date: October 7, 2013*

*Graduation Ceremony Period: November 28, 2013*

*Information Engineering Department*

*Faculty of Sciences and Technology*

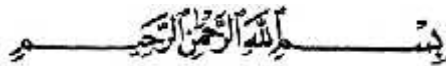
*State Islamic University of Sultan Syarif Kasim Riau*

## ***ABSTRACT***

*The planning and scheduling is a stage that most determine the success of a project. This is because the scheduling dependencies among these activities is the stage of the building project as a whole. Good problem solving scheduling of a project is one of the success factors in the implementation of the project to be completed on time. The final task is to apply genetic algorithms to solve optimization problems in project scheduling. Genetic algorithms represent candidate scheduling solutions into chromosomes at random, and then evaluated using the fitness function and so on are selected. Selection methods used in this study is the roulette wheel selection method, then do crossovers and mutations. At each generation, the chromosomes are evaluated based on the value of the fitness function. After several generations of the genetic algorithm will produce the best chromosome, which is the optimal solution. Implementation of the system in this study using the Hypertext Preprocessing ( php ) and mysql. Testing was conducted on the Blackbox testing, performance, and user acceptance test. Results of the project scheduling system using genetic algorithm is a schedule of activities in a project that could be an alternative decision to the contractor in the execution of the project. And based on the results of performance testing is performed 10 times it can be concluded that the genetic algorithm takes a long time if the value of the iteration, because in this algorithm are the generation process.*

**Key words:** *Crossover, Fitness Function, Genetic Algorithm, Optimization, Project Scheduling, Mutation.*

## KATA PENGANTAR



*Alhamdulillah Rabbil'alam*, penulis bersyukur ke-hadirat Allah SWT, karena atas segala limpahan rahmat dan karunia-Nya yang diberikan sehingga penulis dapat menyelesaikan penelitian dan penulisan laporan tugas akhir ini. *Allahumma sholli'ala Muhammad wa'ala ali sayyidina Muhammad*, yang tidak lupa penulis haturkan juga untuk Rasul Allah, Muhammad SAW.

Laporan tugas akhir ini merupakan salah satu prasyarat untuk memenuhi persyaratan akademis dalam rangka meraih gelar kesarjanaan di Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau (UIN SUSKA Riau). Selama menyelesaikan tugas akhir ini, penulis telah banyak mendapatkan bantuan, bimbingan, dan petunjuk dari banyak pihak baik secara langsung maupun tidak langsung. Untuk itu dalam kesempatan ini penulis ingin mengucapkan terimakasih yang sebesar-besarnya kepada:

1. Prof. Dr. H. M. Nazir, selaku Rektor Universitas Islam Negeri Sultan Syarif Kasim Riau.
2. Dra. Hj. Yenita Morena, M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau.
3. Elin Haerani, ST, M.Kom, selaku Ketua Jurusan Teknik Informatika, Fakultas Sains dan Teknologi.
4. Dr. Okfalisa, ST, MSc, selaku dosen pembimbing tugas akhir. Terimakasih buk untuk waktu yang selalu ibuk luangkan untuk penulis, ilmu, semangat, dan motivasinya yang luar biasa. Terimakasih banyak buk.
5. Elin Haerani, ST, M.Kom, selaku dosen penguji 1 yang banyak membantu dan memberi masukan penulis dalam penyempurnaan Laporan Tugas Akhir ini, terimakasih buk untuk waktu, saran-saran, dan ilmu-ilmunya.

6. Elvia Budianita, ST, M.Cs selaku dosen penguji 2, yang juga banyak membantu dan memberi masukan dalam penyempurnaan Laporan Tugas Akhir ini, terimakasih buk untuk waktu, saran-saran, dan ilmu-ilmunya.
7. Muhammad Affandes, ST, MT, sebagai koordinator tugas akhir yang telah memberi masukan-masukan untuk penyelesaian tugas akhir ini, dan sangat sabar membantu penulis dalam mempersiapkan semua kebutuhan penulis dalam penyelesaian Tugas Akhir ini.
8. Kedua orang tua beserta adik-adikku yang menjadi sumber semangat penulis, atas segenap do'a yang tiada hentinya dan dukungan mereka selama masa Tugas Akhir ini.
9. Efendi Zis, ST, Hendri Kurnia, ST, beserta staff-staff Dinas Prasarana Jalan, Tata Ruang dan Permukiman Provinsi Sumatera Barat yang telah banyak membantu semua kebutuhan untuk Laporan Tugas Akhir ini.
10. Isdarwiliyus, ST, selaku Direktur CV. Kambang Putra.
11. Addurun Nafis R, seseorang yang dengan tulus selalu ada memberikan semangat.
12. Teman-teman karib yang paling baik dan tulus membantu penulis selama ini, Fauzi Azis, Regiolina Hayami, Zulfa Misnur, Sri Suci Giana, Anita, Adrian, dan teman-teman satu angkatan 2008 Teknik Informatika serta pihak-pihak lain yang tidak dapat penulis sebutkan satu persatu. Terimakasih atas saran, bantuan dan do'a serta motivasinya.

Akhirnya, penulis menyadari dalam penulisan laporan ini masih terdapat kekurangan. Oleh karena itu, saran dan kritik sangat penulis harapkan untuk kemajuan penulis secara pribadi. Terimakasih.

Pekanbaru, 7 Oktober 2013

Penulis

## DAFTAR ISI

	Halaman
LEMBAR PERSETUJUAN.....	ii
LEMBAR PENGESAHAN .....	iii
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL.....	iv
LEMBAR PERNYATAAN.....	v
LEMBAR PERSEMBAHAN .....	vi
ABSTRAK.....	vii
<i>ABSTRACT</i> .....	viii
KATA PENGANTAR .....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR .....	xiv
DAFTAR TABEL.....	xvi
DAFTAR RUMUS .....	xviii
DAFTAR LAMPIRAN.....	xix
DAFTAR SIMBOL.....	xx
BAB I PENDAHULUAN .....	I-1
1.1 Latar Belakang.....	I-1
1.2 Rumusan Masalah .....	I-3
1.3 Batasan Masalah.....	I-3
1.4 Tujuan Penelitian.....	I-3
1.5 Sistematika Penulisan.....	I-4
BAB II LANDASAN TEORI.....	II-1
2.1 Konsep Dasar Sistem.....	II-1
2.1.1 Definisi Sistem.....	II-1
2.1.2 Karakteristik Sistem.....	II-1
2.1.3 Komponen Sistem Informasi .....	II-3
2.2 Algoritma Genetika .....	II-3



2.2.1	Struktur Umum Algoritma Genetika .....	II-5
2.2.2	Istilah dalam Algoritma Genetika.....	II-5
2.2.3	Komponen-Komponen Algoritma Genetika.....	II-6
2.2.4	Siklus Umum Algoritma Genetika .....	II-13
2.3	Penjadwalan Proyek .....	II-14
2.4	Proyek.....	II-16
2.4.1	Macam-macam Proyek .....	II-17
2.4.2	Tahapan Pengerjaan Proyek .....	II-18
2.5	Algoritma Genetika dalam Penjadwalan .....	II-19
BAB III METODOLOGI PENELITIAN.....		III-1
3.1	Pengumpulan Data.....	III-2
3.2	Analisa Sistem .....	III-3
3.3	Perancangan Sistem.....	III-4
3.4	Implementasi .....	III-5
3.5	Pengujian .....	III-5
3.6	Kesimpulan dan Saran.....	III-6
BAB IV ANALISIS DAN PERANCANGAN .....		IV-1
4.1	Gambaran Umum Sistem .....	IV-1
4.1.1	Analisa Data Masukan ( <i>Input</i> ).....	IV-2
4.1.2	Analisa Algoritma Genetika dalam Penjadwalan .....	IV-3
4.1.3	Analisa Data Keluaran ( <i>Output</i> ) .....	IV-15
4.1.4	Analisa Kebutuhan Fungsi.....	IV-15
4.2	Perancangan Sistem.....	IV-16
4.2.1	<i>Flowchart</i> Sistem Yang Dibangun .....	IV-17
4.2.2	<i>Context Diagram</i> .....	IV-17
4.2.3	<i>Data Flow Diagram</i> (DFD) .....	IV-18
4.2.4	<i>Entity Relationship Diagram</i> (ERD) .....	IV-22
4.3	Perancangan Basis Data .....	IV-23
4.4	Perancangan Antar Muka ( <i>Interface</i> ) .....	IV-25
4.4.1	Rancangan Struktur Menu .....	IV-25
4.4.2	Rancangan Tampilan Menu Utama .....	IV-25

4.4.3	Rancangan Tampilan Submenu Proyek.....	IV-26
4.4.4	Rancangan Tampilan Submenu Kegiatan.....	IV-27
4.4.5	Rancangan Tampilan Proses Penjadwalan Proyek .....	IV-28
BAB V IMPLEMENTASI DAN PENGUJIAN .....		V-1
5.1	Tahapan Implementasi.....	V-1
5.1.1	Lingkungan Operasional.....	V-1
5.1.2	Implementasi Antar Muka Sistem .....	V-2
5.2	Pengujian Sistem .....	V-11
5.2.1	Pengujian <i>BlackBox</i> .....	V-11
5.2.2	Pengujian Performansi.....	V-14
5.2.3	<i>User Acceptance Test</i> .....	V-16
5.2.4	Hasil Pengujian .....	V-17
5.2.5	Kesimpulan Pengujian .....	V-17
BAB VI PENUTUP.....		VI-1
6.1	Kesimpulan.....	VI-1
6.2	Saran .....	VI-1
DAFTAR PUSTAKA		
LAMPIRAN		
DAFTAR RIWAYAT HIDUP		

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Pembangunan merupakan salah satu usaha pemerintah dalam meningkatkan taraf hidup masyarakat. Tujuannya adalah untuk memberikan kemudahan kepada masyarakat dalam melaksanakan aktifitasnya. Contoh dari pembangunan tersebut adalah pembangunan jalan, jembatan, jaringan telekomunikasi, dan lain-lain. Pembangunan seperti ini biasanya disebut sebagai proyek.

Pada manajemen proyek, sebelum proyek dilaksanakan perlu adanya tahapan pengelolaan yang meliputi perencanaan, penjadwalan, dan pengkoordinasian. Dari ketiga tahapan ini, tahap perencanaan dan penjadwalan merupakan tahapan yang paling menentukan keberhasilan suatu proyek. Hal ini dikarenakan penjadwalan adalah tahap ketergantungan antar aktivitas yang membangun proyek secara keseluruhan. Penjadwalan proyek dengan jumlah aktifitas atau kegiatan yang banyak merupakan tugas yang rumit dan kompleks. Pemecahan masalah penjadwalan yang baik merupakan salah satu faktor keberhasilan dalam pelaksanaan proyek untuk selesai tepat pada waktunya yang merupakan tujuan pokok dan utama, baik bagi kontraktor maupun pemiliknya (Arifudin,2011).

Hal ini juga dialami oleh CV. Kambang Putra sebagai salah satu Perseroan Komanditer yang bergerak dibidang penyediaan jasa kontraktor, yang bekerja dibawah naungan Dinas Prasarana Jalan, Tata Ruang, dan Permukiman Provinsi Sumatera Barat. Selama ini, pembuatan jadwal proyek menggunakan Microsoft Office Excel yang mana penginputan dan analisa pengalokasian waktu setiap kegiatan dihitung secara manual, sehingga jadwal yang dihasilkan tidak optimal dan proses pembuatan jadwal berlangsung lama karena membutuhkan ketelitian. Oleh sebab itu dibutuhkan suatu sistem terkomputerisasi dengan metode yang

tepat untuk membantu *project manager* dalam membuat penjadwalan proyek dan menunjukkan kepada organisasi bagaimana proyek akan dilaksanakan.

Metode – metode yang sudah pernah digunakan untuk penjadwalan seperti *Program Evaluation and Review Technique* (PERT), dan *Critical Path Method* (CPM), dan baru-baru ini banyak peneliti menggunakan pendekatan metode metaheuristik seperti algoritma genetika dan *ant colony optimization*. Penelitian ini akan menggunakan algoritma genetika untuk melakukan penjadwalan terhadap kegiatan dalam suatu proyek. Algoritma genetika adalah algoritma pencarian (*search algorithm*) yang menggunakan prinsip seleksi alam dalam ilmu genetika untuk mengembangkan solusi terhadap permasalahan. Konsep dasar algoritma genetika adalah mengelola suatu populasi individu yang merepresentasikan kandidat solusi sebuah penjadwalan (Arifudin, 2011). Solusi-solusi tersebut dievaluasi menggunakan fungsi *fitness*, dan seterusnya dilakukan seleksi, pindah silang (*crossover*), dan mutasi. Pada penelitian ini metode seleksi yang digunakan yaitu seleksi roda roulette (*Roulette-wheel Selection*) dan pindah silang dilakukan dengan penyilangan satu titik (*One-point Crossover*).

Algoritma genetika ini diusulkan dengan salah satu tujuan untuk mempermudah proses pengalokasian waktu untuk setiap kegiatan dalam sebuah proyek sesuai aturan atau keadaan tertentu, karena setiap masalah yang berbentuk adaptasi dapat diformulasikan dalam terminologi genetis dan bagi proyek dengan kombinasi permasalahan yang cukup kompleks, konsep matematis murni tidak mampu lagi mengakomodasi pencarian solusi permasalahan, sehingga harus menggunakan metode yang cocok untuk mendapatkan solusi penjadwalan yang optimal.

Berbagai penelitian terhadap masalah penjadwalan menggunakan algoritma genetika telah banyak dipelajari dan dikembangkan oleh beberapa penelitian diantaranya : Naso *et al* (2006) membuat penelitian tentang *Genetic Algorithms for supply-chain scheduling (A case study in the distribution of ready-mixed concrete)* menghasilkan solusi yang memuaskan sesuai yang diharapkan, Afandi (2009) menerapkan algoritma genetika untuk masalah penjadwalan *job shop* pada lingkungan industri pakaian, Yendrika (2009) membuat aplikasi

penjadwalan perkuliahan menggunakan algoritma genetika yang menghasilkan jadwal yang optimal yaitu memenuhi semua constraint, dan Riza (2011) menggunakan kombinasi CPM dan algoritma genetika untuk optimasi penjadwalan proyek dengan penyeimbangan biaya, dimana algoritma genetika dapat digunakan untuk melakukan optimasi terhadap pembuatan jadwal proyek.

Berdasarkan uraian latar belakang diatas perlu dibuat sistem penjadwalan proyek dengan metode yang tepat untuk membantu manajer proyek menyusun sebuah jadwal proyek. Jadi dengan menggunakan algoritma genetika diharapkan permasalahan penjadwalan proyek dapat diatasi. Sehingga memberikan kemudahan bagi manajer proyek dalam pembuatan jadwal proyek.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang telah diuraikan di atas, maka dapat diambil suatu rumusan masalah yaitu bagaimana membangun suatu sistem penjadwalan proyek menggunakan algoritma genetika.

## **1.3 Batasan Masalah**

Batasan dalam Tugas Akhir ini :

- a. Proyek-proyek yang digunakan dalam penelitian ini dibatasi pada proyek *engineering-konstruksi*.
- b. Sistem penjadwalan proyek ini dibangun untuk penjadwalan proyek di CV. Kambang Putra.

## **1.4 Tujuan Penelitian**

Tujuan dari penulisan Tugas Akhir ini adalah :

1. Membangun sistem penjadwalan proyek menggunakan algoritma genetika.
2. Menghasilkan output berupa jadwal yang optimal yaitu memenuhi batasan atau persyaratan (*constraints*) dalam penjadwalan proyek, meliputi beberapa kegiatan tidak bisa dimulai sebelum kegiatan yang lain selesai, mengoptimalkan efisiensi pemakaian waktu pelaksanaan

setiap kegiatan, dan tidak boleh melebihi durasi proyek yang telah ditetapkan.

### **1.5 Sistematika Penulisan**

Sistematika dalam penulisan Tugas Akhir ini antara lain :

#### **Bab I Pendahuluan**

Menjelaskan latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, dan sistematika penulisan laporan Tugas Akhir.

#### **Bab II Landasan Teori**

Menjelaskan teori – teori tentang sistem, algoritma genetika, dan penjadwalan proyek.

#### **Bab III Metodologi Penelitian**

Menjelaskan tentang tahap – tahap pengerjaan Tugas Akhir.

#### **Bab IV Analisa Sistem dan Perancangan**

Menjelaskan tentang gambaran umum sistem, analisa sistem yang akan dibuat, dan perancangan sistem.

#### **Bab V Implementasi dan Pengujian**

Menjelaskan implementasi sistem yang meliputi lingkungan implementasi dan implementasi antarmuka (*interface*) sistem serta pengujian sistem.

#### **Bab VI Penutup**

Bab ini berisikan kesimpulan dan saran – saran yang berkaitan dengan penelitian ini.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Konsep Dasar Sistem**

Sistem merupakan kumpulan elemen-elemen yang saling terkait dan bekerjasama untuk memproses masukan (*input*) yang ditujukan kepada sistem tersebut dan mengolah masukan tersebut sampai menghasilkan keluaran (*output*) yang diinginkan.

##### **2.1.1 Definisi Sistem**

Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu (Jogiyanto, 2005). Secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel-variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu.

Dalam mendefinisikan sistem, terdapat dua kelompok pendekatan sistem yang berbeda, yaitu (Jogiyanto, 1990):

- a. Pendekatan sistem pada prosedural.

Mendefinisikan sistem sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu.

- b. Pendekatan sistem yang menekankan pada elemen atau komponen.

Mendefinisikan sistem sebagai suatu kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.

##### **2.1.2 Karakteristik Sistem**

Adapun karakteristik dari suatu sistem adalah sebagai berikut (Jogiyanto, 1990):

a. Bagian (*Component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi yang artinya bekerja sama membentuk satu kesatuan komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu sub sistem atau bagian dari sistem.

b. Batas Sistem (*Boundary*)

Merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luar.

c. Lingkungan Luar Sistem (*Environment*)

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem.

d. Penghubung Sistem (*Interface*)

Penghubung (*interface*) merupakan media penghubung antara satu sub sistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber daya yang mengalir dari suatu subsistem akan menjadi masukan untuk subsistem yang lainnya melalui penghubung ini, dengan begitu suatu sistem dapat berintegrasi dengan subsistem yang lainnya dengan membentuk suatu kesatuan.

e. Masukan Sistem (*Input*)

Masukan adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa maintenance input dan signal input. Maintenance input adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Signal input adalah energi yang diproses untuk diperoleh keluarannya.

f. Keluaran Sistem (*Output*)

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan keluaran dapat merupakan masukan untuk sistem yang lain atau supra sistem.

g. Pengolah Sistem (*Process*)

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran. Suatu sistem pengolahan akan mengolah



berupa bahan baku dalam hal ini adalah data serta bahan-bahan lainnya untuk menghasilkan keluaran berupa laporan dan informasi yang berguna.

h. Sasaran Sistem (*Objective*) atau tujuan (*Goal*)

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Jika suatu sistem tidak mempunyai sasaran maka operasi tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali masukkan yang dibutuhkan dan keluaran sistem yang akan dihasilkan oleh sistem itu sendiri.

### 2.1.3 Komponen Sistem Informasi

Secara rinci komponen-komponen sistem informasi dapat dijelaskan sebagai berikut (Kadir, 2003):

- a. Perangkat keras (*hardware*), mencakup piranti-piranti fisik seperti monitor dan printer.
- b. Perangkat lunak (*software*) atau program, yaitu sekumpulan intruksi yang memungkinkan perangkat keras untuk dapat memproses data.
- c. Prosedur, yaitu sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan menghasilkan keluaran yang diinginkan.
- d. Pengguna (*user*), yaitu semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan, dan penggunaan keluaran sistem informasi.
- e. *Database*, merupakan kumpulan dari data yang saling berhubungan dengan data lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya, diantaranya data *user* dan sistem.

Komponen-komponen diatas sangat penting dalam suatu sistem informasi, apabila salah satu komponen tidak ada maka sistem informasi tidak akan berjalan dengan baik.

## 2.2 Algoritma Genetika

Algoritma genetika adalah algoritma pencarian/heuristik yang didasarkan atas mekanisme seleksi alam dan evolusi biologis. Algoritma genetika pertama kali

diperkenalkan oleh John Holland dalam bukunya yang berjudul “*Adaption in natural and artificial systems*”, dan oleh De Jong dalam bukunya “*Adaption of the behavior of a class of genetic adaptive systems*”, yang keduanya diterbitkan pada tahun 1975, yang merupakan dasar dari algoritma genetika (Davis, 1991 dikutip dari Putra, 2009).

Algoritma genetika berbeda dengan teknik pencarian konvensional, dimana pada algoritma genetika kondisi diawali dengan setting awal solusi acak yang disebut populasi. Tiap individu dalam populasi disebut kromosom, yang merepresentasikan suatu solusi atas permasalahan. Kromosom berevolusi melalui iterasi berkelanjutan, yang disebut generasi. Selama tiap generasi, kromosom dievaluasi menggunakan beberapa ukuran *fitness*. Untuk menghasilkan generasi berikutnya, kromosom baru yang disebut *offspring*, dibentuk baik melalui penyatuan dua kromosom dari generasi awal menggunakan operator perkawinan silang (*crossover*) atau memodifikasi kromosom menggunakan operator mutasi (*mutation*). Suatu generasi baru dibentuk melalui proses seleksi beberapa induk (*parents*) dan anak (*offspring*), sesuai dengan nilai *fitness*, dan melalui eliminasi kromosom lainnya agar ukuran populasi tetap konstan. Kromosom yang sesuai memiliki kemungkinan tertinggi untuk dipilih. Setelah beberapa generasi, algoritma menghasilkan kromosom-kromosom terbaik yang diharapkan mewakili solusi optimal atau sub optimal atas permasalahan. (Susetyo, 2004 dikutip dari Yendrika Putra, 2009)

Menurut Suyanto (2005) algoritma genetika telah banyak diaplikasikan untuk penyelesaian masalah dan pemodelan dalam bidang teknologi, bisnis, dan *entertainment*, seperti : optimasi, pemrograman otomatis, *machine learning*, model ekonomi, model sistem imunisasi, model ekologis, dan interaksi antara evolusi dan belajar.

Ada 3 keuntungan utama dalam mengaplikasikan algoritma genetika pada masalah-masalah optimasi (Sam’ani, 2012) :

- a. Algoritma genetika tidak memerlukan kebutuhan matematis banyak mengenai masalah optimasi.

- b. Kemudahan dan kenyamanan pada operator-operator evolusi membuat algoritma genetika sangat efektif dalam melakukan pencarian global.
- c. Algoritma genetika menyediakan banyak fleksibilitas untuk digabungkan dengan metode heuristik yang tergantung domain, untuk membuat implementasi yang efisien pada masalah-masalah khusus.

### 2.2.1 Struktur Umum Algoritma Genetika

Algoritma genetika memberikan suatu pilihan bagi penentuan nilai parameter dengan meniru cara reproduksi genetik, pembentukan kromosom baru serta seleksi alami seperti terjadi pada makhluk hidup.

Inisialisasi populasi awal dilakukan untuk menghasilkan solusi awal dari suatu permasalahan algoritma genetika. Inisialisasi ini dilakukan secara acak sebanyak jumlah kromosom yang diinginkan. Selanjutnya dihitung nilai fitness dan seterusnya dilakukan seleksi dengan menggunakan metode roda roulette, tournament atau ranking. Kemudian dilakukan perkawinan silang (*crossover*) dan mutasi. Setelah melalui beberapa generasi maka algoritma ini akan berhenti sebanyak generasi yang diinginkan. Sebagaimana halnya proses evolusi di alam, suatu algoritma genetika yang sederhana umumnya terdiri dari tiga operator yaitu: operator reproduksi, operator *crossover* (persilangan) dan operator mutasi.

Ada dua hal penting yang harus dilakukan pada awal proses Algoritma Genetika. Pertama, pendefinisian atau pengkodean kromosom yang merupakan solusi yang masih berbentuk simbol. Kedua, penentuan fungsi *fitness* atau fungsi obyektif. Dua hal ini berperan penting dalam algoritma genetika untuk menyelesaikan suatu masalah (Arifudin,2011).

### 2.2.2 Istilah dalam Algoritma Genetika

Terdapat beberapa definisi penting dalam algoritma genetika yang perlu diperhatikan, yaitu :

- a. *Genotype* (gen), merupakan sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu keasatuan gen yang

dinamakan kromosom. Gen mengkodekan informasi yang disimpan di dalam kromosom.

- b. *Allele*, merupakan nilai dari gen.
- c. Individu atau kromosom, gabungan gen-gen yang membentuk nilai tertentu dan merupakan suatu alternatif solusi terhadap permasalahan yang akan diselesaikan.
- d. Populasi, merupakan sekumpulan individu yang akan diproses bersama dalam satu siklus proses evaluasi.
- e. Generasi, merupakan satu siklus proses evolusi atau satu iterasi di dalam algoritma genetika.

### 2.2.3 Komponen – Komponen Algoritma Genetika

Ada beberapa komponen dalam algoritma genetika, yaitu :

#### 1. Pengkodean

Teknik pengkodean adalah bagaimana mengkodekan gen dari kromosom, gen merupakan bagian dari kromosom. Satu gen akan mewakili satu variabel. Agar dapat diproses melalui algoritma genetika, maka alternatif solusi tersebut harus dikodekan terlebih dahulu kedalam bentuk kromosom. Masing-masing kromosom berisi sejumlah gen yang mengodekan informasi yang disimpan didalam individu atau kromosom.

Berdasarkan jenis simbol yang digunakan sebagai nilai suatu gen, metode pengkodean dapat diklasifikasikan sebagai berikut (Arifudin, 2011):

- a. Pengkodean biner merupakan cara pengkodean yang paling umum digunakan, karena pengkodean ini merupakan yang pertama kali digunakan dalam algoritma genetika oleh Holland (Cordon et al, 2001). Pengkodean biner dinyatakan dalam kromosom biner.
- b. Pengkodean bilangan bulat. Pengkodean ini baik digunakan untuk masalah optimasi kombinatorial. Dengan pengkodean bilangan bulat, ukuran kromosom menjadi lebih sederhana dan tidak terlalu panjang.

- c. Pengkodean bilangan riil. Gen dan Cheng (2000) menyatakan bahwa pengkodean bilangan riil baik digunakan untuk masalah optimasi fungsi dan optimasi kendala.
- d. Pengkodean struktur data adalah model pengkodean yang menggunakan struktur data. Pengkodean ini digunakan untuk masalah yang lebih kompleks seperti perencanaan trajektori robot dan masalah pewarnaan grap.

### Inisialisasi Populasi Awal

Inisialisasi populasi awal merupakan suatu metode untuk menghasilkan kromosom-kromosom awal. Jumlah individu pada populasi awal merupakan masukan dari pengguna. Setelah jumlah individu pada populasi awal ditentukan, dilakukan inisialisasi terhadap kromosom yang terdapat pada populasi tersebut. Inisialisasi dilakukan secara acak, namun tetap memperhatikan domain solusi dan kendala permasalahan yang ada.

### 2. Fungsi Evaluasi (Fungsi *Fitness*)

Fungsi evaluasi dalam algoritma genetika merupakan sebuah fungsi yang memberikan penilaian kepada kromosom (*fitness value*) untuk dijadikan suatu acuan dalam mencapai nilai optimal pada algoritma genetika. Nilai fitness ini kemudian menjadi nilai bobot suatu kromosom. Ada dua hal yang harus dilakukan dalam melakukan evaluasi kromosom, yaitu: evaluasi fungsi objektif (fungsi tujuan) dan konversi fungsi objektif ke dalam fungsi *fitness*. Secara umum fungsi fitness ditentukan dari fungsi objektif dengan nilai yang tidak negatif, jika ternyata nilai dari fungsi objektif bernilai negatif maka perlu ditambahkan suatu konstanta  $x$  agar nilai *fitness* yang terbentuk tidak bernilai negatif (Putra, 2009).

Didalam evolusi alam, individu yang bernilai *fitness* tinggi yang akan bertahan hidup. Sedangkan individu yang bernilai *fitness* rendah akan mati. Pada masalah optimasi, jika solusi yang akan dicari adalah memaksimalkan fungsi  $h$  ( dikenal sebagai masalah maksimasi ) sehingga nilai *fitness* yang digunakan adalah nilai dari fungsi  $h$  tersebut, yakni  $f = h$  (dimana  $f$  adalah nilai *fitness*). Tetapi jika masalahnya adalah meminimalkan fungsi  $h$  (masalah minimasi), maka fungsi  $h$

tidak bisa digunakan secara langsung. Hal ini disebabkan adanya aturan bahwa individu yang memiliki nilai *fitness* tinggi lebih mampu bertahan hidup pada generasi berikutnya. Oleh karena itu nilai *fitness* yang bisa digunakan adalah  $f = 1/h$ , yang artinya semakin kecil nilai  $h$ , semakin besar nilai  $f$ . Tetapi hal ini akan menjadi masalah jika  $h$  bisa bernilai 0, yang mengakibatkan  $f$  bisa bernilai tak hingga. Untuk mengatasinya,  $h$  perlu ditambah sebuah bilangan yang dianggap kecil [0-1] sehingga nilai *fitness*nya menjadi :

$$f = 1/h + a \dots\dots\dots \text{(Persamaan 2.1)}$$

dengan  $a$  adalah bilangan yang kecil dan bervariasi [0-1] sesuai dengan masalah yang akan diselesaikan (Suyanto,2005). Oleh karena itu fungsi *fitness* menjadi masalah atau penentu utama keberhasilan algoritma genetika. Didalam penelitian ini batasan atau *constraint* dalam penyusunan jadwal proyek yang dijadikan fungsi objektifnya yaitu meminimumkan pelanggaran terhadap *constraint* yang telah ditentukan. Jadi persamaan (2.1) cocok dipakai pada kasus dalam penelitian ini.

### 3. Seleksi

Seleksi merupakan proses pemilihan orang tua untuk reproduksi (biasanya didasarkan pada nilai *fitness*). Seleksi bertujuan untuk memberikan kesempatan reproduksi yang paling besar bagi anggota populasi yang paling baik. Ada beberapa metode yang bisa digunakan dalam tahap seleksi, diantaranya :

#### a. *Range Base Fitness*

Populasi diurutkan berdasarkan nilai objektifnya. Nilai *fitness* dari tiap-tiap individu hanya tergantung pada posisi individu tersebut dalam urutan dan tidak dipengaruhi oleh nilai objektifnya.

#### b. *Roulette Wheel Selection* (Seleksi Roda Roulette)

Individu-individu dipetakan dalam suatu segmen garis secara berurutan sedemikian hingga tiap-tiap segmen individu memiliki urutan yang sama dengan ukuran *fitness*. Sebuah bilangan acak dibangkitkan dan individu yang memiliki segmen dalam kawasan bilangan random tersebut akan terseleksi. Cara penyelesaian metode ini meniru permainan roda *roulette*, dimana prosedur seleksi dimulai dengan memutar roda *roulette* sebanyak

n, setiap waktu dipilih satu kromosom sebagai induk untuk menghasilkan kromosom baru. Metode *roulette-wheel selection* sangat mudah diimplementasikan dalam pemrograman. Pertama, dibuat interval nilai kumulatif (dalam interval  $[0,1]$ ) dari nilai *fitness* masing-masing kromosom dibagi total nilai *fitness* dari semua kromosom. Sebuah kromosom akan terpilih jika bilangan random yang dibangkitkan berada dalam interval akumulatifnya. Metode ini merupakan salah satu yang paling umum digunakan (Goldsberg, 1989 dikutip dari Riza Arifudin, 2011).

c. *Stochastic Universal Sampling*

Individu-individu dipetakan dalam suatu segmen garis secara berurutan sedemikian hingga hingga tiap-tiap segmen individu memiliki urutan yang sama dengan ukuran *fitness*. Kemudian diberikan sejumlah pointer sebanyak individu yang ingin diseleksi pada garis tersebut.

d. *Local Selection* (Seleksi Lokal)

Tiap individu yang berada pada konstrain tertentu disebut nama lingkungan lokal. Interaksi antar individu hanya dilakukan di dalam wilayah tersebut. Lingkaran tersebut ditetapkan sebagai struktur dimana populasi tersebut terdistribusi.

e. *Truncation Selection* (Seleksi dengan pemotongan)

Individu diurutkan berdasarkan *fitness*, hanya individu yang terbaik saja yang akan diseleksi sebagai induk. Parameter yang digunakan dalam metode ini adalah suatu nilai ambang (*trunk*) yang mengidentifikasi ukuran populasi yang akan diseleksi sebagai induk yang berkisar antara 50%-100%.

f. *Tournament Selection* (Seleksi dengan turnamen)

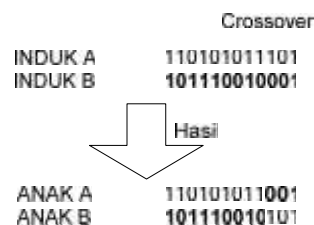
Dalam bentuk paling sederhana, metode ini mengambil dua kromosom secara random dan kemudian menyeleksi salah satu yang bernilai *fitness* paling tinggi untuk menjadi orang tua pertama. Cara yang sama dilakukan lagi untuk mendapatkan orang tua yang kedua. Metode *tournament selection* yang lebih rumit adalah dengan mengambil  $m$  kromosom secara

random. Kemudian kromosom bernilai *fitness* tertinggi dipilih sebagai orang tua pertama jika bilangan random yang dibangkitkan kurang dari suatu nilai batas yang ditentukan  $p$  dalam interval  $[0,1]$ . Pemilihan orang tua akan dilakukan secara random dari  $m - 1$  kromosom yang ada jika bilangan random yang dibangkitkan lebih dari atau sama dengan  $p$ . Pada *tournament selection*, variabel  $m$  adalah *tournament size* dan  $p$  adalah *tournament probability*. Biasanya  $m$  diset sebagai suatu nilai yang sangat kecil, misal 4 atau 5. Sedangkan  $p$  biasanya diset sekitar 0,75. (Suyanto, 2005)

#### 4. Pindah Silang (*Crossover*)

Pindah silang atau *crossover* adalah sebuah proses yang membentuk kromosom baru dari dua kromosom induk dengan menggabungkan bagian informasi dari masing-masing kromosom. *Crossover* menghasilkan kromosom baru yang disebut kromosom anak (*offspring*). *Crossover* bertujuan untuk menambah keanekaragaman string dalam satu populasi dengan penyilangan antar string yang diperoleh dari reproduksi sebelumnya (Arifudin, 2011).

*Crossover* merupakan operator genetik utama, yang beroperasi pada dua kromosom dalam suatu waktu dan menghasilkan *offspring* dengan mengkombinasikan kedua fitur-fitur kromosom (Fadlisyah dkk, 2009). Sebuah kromosom yang mengarah pada solusi yang bagus bisa diperoleh dari proses memindah-silangkan dua buah kromosom. Contoh *crossover* dapat dilihat pada gambar berikut:



Gambar 2.1 Proses Pindah Silang (Putra, 2009)

Proses *crossover* dilakukan dengan cara memilih dua induk dengan kualitas yang baik, setelah itu dilakukan proses ekstraksi kromosom setiap induk.



Titik potong ditentukan secara acak, kemudian dilakukan pertukaran bit-bit kromosom disebelah kanan titik kromosom sehingga terbentuk keturunan yaitu anak A dan B. Kromosom anak sebagian besar masih mewarisi kromosom induk tetapi sebagian lagi sudah terjadi pertukaran materi genetik antar kromosom.

Pindah silang bisa juga berakibat buruk jika ukuran populasinya sangat kecil. Dalam suatu populasi yang sangat kecil, suatu kromosom dengan gen-gen yang mengarah ke solusi akan sangat cepat menyebar ke kromosom-kromosom lainnya. Untuk mengatasi masalah ini digunakan suatu aturan bahwa pindah silang hanya bisa dilakukan dengan suatu probabilitas tertentu (probabilitas *crossover*). Artinya pindah silang bisa dilakukan hanya jika suatu bilangan random  $[0,1]$  yang dibangkitkan kurang dari probabilitas *crossover* ( $P_c$ ) yang ditentukan. Pada umumnya  $P_c$  diset mendekati 1, misalnya 0.8 (Suyanto,2005). Probabilitas *crossover* ( $P_c$ ) bertujuan untuk mengendalikan operator *crossover*. Jika  $n$  adalah banyaknya string pada populasi, maka sebanyak  $(P_c) \times n$  string akan mengalami *crossover*. Semakin besar nilai ( $P_c$ ), semakin cepat pula string baru muncul dalam populasi. Dan juga jika ( $P_c$ ) terlalu besar, string yang merupakan kandidat solusi terbaik mungkin dapat hilang lebih cepat pada generasi berikutnya.

Proses *crossover* memiliki nilai kemungkinan yang besar dalam satu siklus algoritma genetika karena tujuan utamanya adalah membentuk keragaman individu, semakin tinggi probabilitas *crossover* maka semakin cepat keragaman terbentuk. *Crossover* dapat dilakukan dengan beberapa cara yang berbeda, diantaranya:

a. Penyilangan Satu Titik (*One-point Crossover*)

Metode *crossover* ini yang sering digunakan pada algoritma genetika. Pada penyilangan satu titik, posisi penyilangan  $k$  ( $k = 1,2,3,...,n$ ) dengan  $n$ = panjang kromosom yang diseleksi secara acak. Pada titik tersebut dilakukan pertukaran antar kromosom induk untuk menghasilkan anak.

b. Penyilangan Banyak Titik (*Multi-point Crossover*)

Untuk kromosom yang sangat panjang, misalkan 1000 gen, mungkin saja diperlukan beberapa titik potong. Penyilangan dilakukan sebanyak  $m$  ( $m = 1, 2, 3,...,n$ ) dengan posisi penyilangan  $k$  ( $k = 1,2,3,...,n$ ) yang ditentukan

secara acak. Pada titik tersebut dilakukan pertukaran antar kromosom induk untuk menghasilkan anak.

c. Penyilangan Seragam (*Uniform Crossover*)

Dibentuk suatu kromosom sepanjang kromosom induk dengan bit-bit yang dipilih secara acak, kemudian penyilangan dilakukan sebanyak  $m$  ( $m = 1, 2, 3, \dots, n$ ) dengan posisi penyilangan  $k$  ( $k = 1, 2, 3, \dots, n$ ) yang ditentukan secara acak. Pada titik tersebut dilakukan pertukaran antar kromosom induk untuk menghasilkan anak.

## 5. Mutasi

Mutasi merupakan proses mengubah secara acak nilai dari satu atau beberapa gen dalam suatu kromosom (Haupt, 2004 dikutip dari Riza Arifudin, 2011). Mutasi adalah operator algoritma genetika yang bertujuan untuk membentuk individu-individu yang baik atau memiliki kualitas diatas rata-rata. Selain itu mutasi dipergunakan untuk mengembalikan kerusakan materi genetik akibat proses *crossover*.

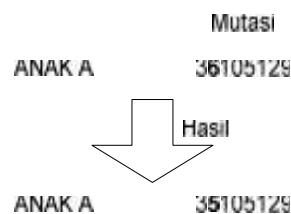
Di dalam algoritma genetika, mutasi melakukan peran vital berikut (Fadlisyah dkk, 2009) :

- a. Menempatkan kembali gen-gen yang hilang dari populasi sepanjang proses seleksi, agar mereka dapat dilibatkan kembali pada konteks yang berikutnya.
- b. Menyediakan gen-gen yang tidak hadir pada populasi awal.

Pada mutasi terdapat satu parameter yang sangat penting, yaitu probabilitas mutasi ( $P_m$ ) yang bertujuan untuk mengendalikan operator mutasi. Probabilitas mutasi didefinisikan sebagai persentasi dari jumlah total gen dalam populasi yang akan mengalami mutasi. Disetiap generasi diperkirakan terjadi mutasi sebanyak  $(P_m) \times n$  sring. Pada seleksi alam murni, mutasi jarang sekali muncul sehingga probabilitas mutasi yang digunakan umumnya kecil, lebih kecil dari probabilitas *crossover*.  $P_m$  biasanya diset antara  $[0-1]$ , misalnya 0.1 (Suyanto, 2005). Misalkan *offspring* yang terbentuk adalah 100 dengan jumlah gen setiap kromosom adalah 4 dan peluang mutasi adalah 0.10, maka diharapkan

terdapat 40 kromosom dari 400 gen yang ada pada populasi tersebut akan mengalami mutasi.

Mutasi yang digunakan dalam penelitian ini adalah mutasi dengan pengkodean bulat. Mutasi gen ini dilakukan dengan cara pemilihan nilai secara acak. Suatu gen yang terpilih untuk dimutasi nilainya diganti dengan nilai baru yang dibangkitkan secara acak dalam interval nilai-nilai gen yang diizinkan. Misalnya, jika nilai-nilai gen berada dalam interval  $[0.9]$ , maka gen baru yang dibangkitkan secara acak juga berada dalam interval  $[0.9]$ . Nilai gen baru yang dihasilkan bisa saja dibatasi dengan aturan berbeda dengan nilai lama.



Gambar 2.2 Proses Mutasi

## 6. *Elitism*

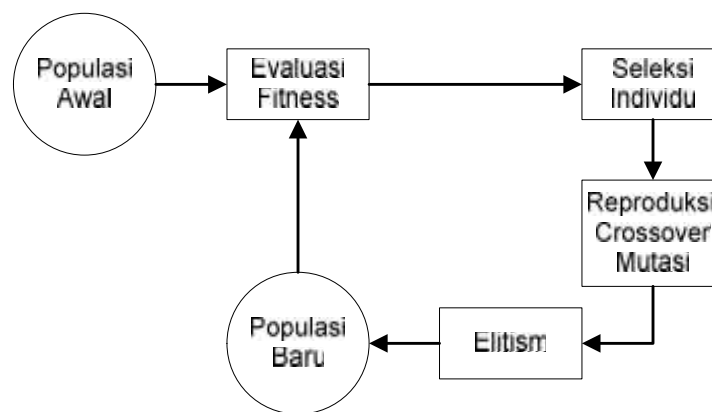
Karena seleksi dilakukan secara acak (*random*), maka tidak ada jaminan bahwa suatu individu bernilai *fitness* tertinggi akan selalu terpilih. Kalaupun individu yang memiliki nilai *fitness* tertinggi terpilih, mungkin saja individu tersebut akan rusak (nilai *fitness* menurun) karena proses *crossover*. Untuk menjaga agar individu yang bernilai *fitness* tertinggi tidak hilang selama evolusi, maka perlu dibuat satu atau beberapa buah duplikatnya. Proses ini dikenal sebagai *elitism*.

### 2.2.4 Siklus Umum Algoritma Genetika

Langkah Umum pada algoritma genetika adalah sebagai berikut (Putra, 2009):

- a. Melakukan inisialisasi populasi kromosom dengan solusi secara acak (*random*).

- b. Melakukan evaluasi setiap kromosom dalam populasi menggunakan persamaan fungsi evaluasi (*fitness function*).
- c. Memilih sebagian anggota populasi sebagai solusi yang sesuai dengan induknya untuk generasi selanjutnya.
- d. Menciptakan solusi (keturunan) baru dengan mengawinkan solusi dari induknya dengan cara *crossover* dan mutasi.
- e. Membuang atau menghapus anggota populasi lama yang tidak produktif untuk membuat ruang solusi yang baru agar dapat masuk kedalam populasi.
- f. Jika aturan pemberhentian terpenuhi, berhenti dan keluarkan kromosom yang paling baik. Jika tidak kembali ke langkah 3.



Gambar 2.3 Siklus Umum Algoritma Genetika.

### 2.3 Penjadwalan Proyek

Penjadwalan merupakan proses atau cara membagi waktu berdasarkan rencana pengaturan urutan kerja, biasanya berupa tabel kegiatan atau rencana kegiatan dengan pembagian waktu pelaksanaan yang terperinci. Penjadwalan diperlukan sebagai tolak ukur dalam melakukan kegiatan/aktifitas tertentu.

Penjadwalan proyek adalah pembuatan rencana pelaksanaan setiap kegiatan di dalam suatu proyek dengan mengoptimalkan efisiensi pemakaian waktu dan sumber daya yang tersedia, tetapi kesesuaian presedensi diantara kegiatan tetap dipenuhi (Arifudin,2011).

Penjadwalan proyek terdiri dari macam-macam kegiatan dalam sebuah proyek serta waktu pelaksanaan tiap-tiap kegiatan. Terdapat batasan/persyaratan (*constraints*) dalam penyusunan penjadwalan proyek. *Constraint* merupakan suatu syarat yang harus terpenuhi dan tidak boleh dilanggar terhadap kendala yang ditetapkan agar dapat menghasilkan susunan penjadwalan yang baik. Beberapa *constraint* tersebut, yaitu :

1. Beberapa kegiatan dilaksanakan setelah kegiatan pendahulunya (*predecessor*) selesai.
2. Durasi setiap kegiatan tidak boleh kurang dari yang telah ditentukan.
3. Tidak boleh melampaui waktu pelaksanaan proyek yang telah ditetapkan.

Jika terjadi pelanggaran terhadap kendala yang ditetapkan maka akan diberikan suatu nilai penalti atau hukuman antara 0 sampai 1 untuk setiap pelanggaran. Semakin kecil jumlah pelanggaran yang terjadi, solusi penjadwalan yang dihasilkan akan semakin baik.

Menurut Lawrence dan Pasternack (2001) (dikutip dari Riza Arifudin, 2011) ada beberapa tujuan penjadwalan proyek meliputi :

- a. Menentukan jadwal paling awal dan paling akhir dari waktu mulai dan berakhir untuk setiap kegiatan yang mengarah ke waktu penyelesaian paling awal untuk keseluruhan proyek.
- b. Menghitung kemungkinan bahwa proyek akan selesai dalam jangka waktu tertentu.
- c. Mencari biaya jadwal minimum yang akan menyelesaikan sebuah proyek dengan tanggal tertentu.
- d. Menginvestigasi bagaimana keterlambatan untuk kegiatan tertentu mempengaruhi waktu penyelesaian keseluruhan proyek.
- e. Monitoring sebuah proyek untuk menentukan apakah berjalan tepat waktu dan sesuai anggaran.
- f. Mencari jadwal kegiatan yang akan memuluskan alokasi sumber daya selama durasi proyek.

## 2.4 Proyek

Menurut *Project Management Institute* (PMI), dalam buku yang berjudul “*A Guide to the Project Management Body of Knowledge*” (PMBOK Guide) mendefinisikan proyek sebagai berikut: “Usaha temporer yang dilakukan untuk menciptakan proyek atau jasa (*service*) yang unik” (Schwalbe, 2004 dikutip dari Maya Erika, 2008). Yang membedakan proyek dengan pekerjaan lain adalah sifatnya yang khusus dan tidak bersifat rutin pengadaannya, sehingga pengelolaannyapun memerlukan perhatian ekstra lebih banyak.

Dari pengertian diatas maka ciri pokok proyek adalah sebagai berikut (Iman, 1999 dikutip dari Maya Erika, 2008):

- a. Bertujuan menghasilkan lingkup (*scope*) tertentu berupa produk akhir atau hasil kerja akhir.
- b. Dalam proses mewujudkan lingkup diatas, ditentukan sejumlah biaya, jadwal, serta kriteria mutu.
- c. Bersifat sementara, dalam arti umurnya dibatasi oleh selesainya tugas. Titik awal dan akhir ditentukan dengan jelas.
- d. Nonrutin, tidak berulang-ulang. Macam dan intensitas kegiatan berubah sepanjang proyek berlangsung.

Sebuah proyek mendefinisikan satu kombinasi kegiatan-kegiatan yang saling berkaitan yang harus dilakukan dalam urutan tertentu sebelum keseluruhan tugas dapat diselesaikan (Arifudin, 2011). Kegiatan-kegiatan ini saling berkaitan dalam satu urutan yang logis dalam arti bahwa beberapa kegiatan tidak dapat dimulai sampai kegiatan-kegiatan lainnya diselesaikan. Sebuah kegiatan (pekerjaan) dalam sebuah proyek biasanya dipandang sebagai sebuah tugas yang memerlukan waktu dan sumber daya untuk penyelesaiannya.

Jadi proyek adalah sembarang pekerjaan yang memiliki awal dan akhir, terdiri dari beberapa pekerjaan/aktivitas yang harus dilaksanakan dengan urutan tertentu. Tiap-tiap pekerjaan membutuhkan sumber daya (biaya, tenaga, waktu) yang berbeda-beda.

### 2.4.1 Macam-macam Proyek

Dilihat dari komponen kegiatan utamanya macam proyek dapat dikelompokkan sebagai berikut (Imam, 1999 dikutip dari Maya Erika, 2008):

- a. *Proyek Engineering-Konstruksi*, komponen utama jenis proyek ini terdiri dari pengkajian kelayakan, desain *engineering*, pengadaan dan konstruksi. Contoh proyek macam ini adalah pembangunan jalan, jembatan, gedung, fasilitas industri, dan lain-lain. Dalam Tugas Akhir ini aplikasi yang dibangun menangani pengelolaan proyek tahap penjadwalan untuk proyek kelompok *engineering-konstruksi*.
- b. *Proyek Engineering-Manufaktur*, merupakan proyek yang dilaksanakan untuk menghasilkan produk baru. Kegiatan utamanya meliputi desain *engineering*, pengembangan produk, pengadaan, manufaktur, perakitan, uji coba fungsi dan operasi produk yang dihasilkan. Contohnya adalah pembuatan ketel uap, generator listrik, mesin pabrik, dan lain-lain.
- c. *Proyek Penelitian dan Pengembangan*, proyek ini bertujuan untuk melakukan penelitian dan pengembangan dalam rangka menghasilkan suatu produk tertentu.
- d. *Proyek Pelayanan Manajemen*, contohnya merancang sistem informasi manajemen meliputi perangkat lunak atau perangkat keras, merancang program efisiensi dan penghematan. Proyek jenis ini tidak menghasilkan produk dalam bentuk fisik, tetapi laporan akhir.
- e. *Proyek Kapital*, berbagai badan usaha atau pemerintah memiliki kriteria tertentu untuk proyek kapital. Proyek kapital umumnya meliputi pembebasan tanah, pembelian material dan lain-lain.
- f. *Proyek Radio-Telekomunikasi*, tujuannya untuk membangun jaringan telekomunikasi yang dapat menjangkau area yang luas dengan biaya yang relatif tidak terlalu mahal.
- g. *Proyek Konservasi Bio-Diversity*, proyek ini berkaitan dengan usaha pelestarian lingkungan.

### 2.4.2 Tahapan Pengerjaan Proyek

Dalam menjalankan sebuah proyek, PMI mengakui lima kategori aktifitas proyek yang umum disebut sebagai “proses proyek” atau tahapan pengerjaan proyek (Scwalbe, 2004 dikutip dari Maya Erika, 2008):

1. Inisialisasi

Inisialisasi merupakan langkah awal proyek. Tahapan ini bertujuan untuk melakukan penilaian terhadap suatu proyek, apakah layak untuk dikerjakan atau tidak. Dalam tahapan ini ditentukan hasil akhir yang akan dicapai, dan melihat proyek sejenis yang pernah dikerjakan sebelumnya (jika ada). Tahapan ini dilakukan oleh manajer proyek dan tim proyek lainnya dalam rapat internal perusahaan.

2. Perencanaan

Perencanaan proyek diperlukan untuk pegangan kegiatan implementasi, komunikasi para pelaksana dengan *stakeholder*, serta dipakai sebagai tolak ukur pengendalian.

Pada tahap ini akan ditentukan beberapa hal seperti batasan pengerjaan, hasil akhir yang ingin dicapai, sumber daya manusia yang terlibat di dalam proyek serta perencanaan jadwal proyek.

3. Pengeksekusian

Pengeksekusian merupakan pelaksanaan segala sesuatu yang merupakan hasil dari tahap perencanaan. Bersamaan dengan pengeksekusian, dilakukan juga tahap pengendalian.

4. Pengendalian

Bersamaan dengan mulai dilakukannya pengeksekusian, maka tahap ini mulai dijalankan dengan tujuan untuk menjaga agar pengeksekusian tetap sesuai dengan rencana. Jika pengeksekusian menyalahi rencana maka tahap pengendalian akan dilakukan. Macam-macam pengendalian yang ada dalam tahap ini antara lain adalah (Imam, 1999 dikutip dari Maya Erika, 2008):

- a. Pengendalian biaya, merupakan proses memantau dan mencatat apakah penggunaan biaya telah sesuai dengan perencanaan. Bila tidak



sesuai, dicari sebabnya dan dievaluasi dampak yang mungkin terjadi serta diadakannya koreksi.

- b. Pengendalian jadwal, meliputi kegiatan yang berkaitan dengan pemantauan dan pengkoreksian agar “*progress*” pekerjaan proyek sesuai dengan jadwal yang telah ditentukan.
- c. Pengendalian mutu, meliputi kegiatan yang berkaitan dengan pemantauan apakah proses dan hasil kerja tertentu proyek tersebut memenuhi standar mutu yang bersangkutan, serta pengidentifikasian cara untuk mencegah terjadinya hasil yang tidak memuaskan.

#### 5. Penyelesaian

Tahapan ini merupakan tahap akhir pengerjaan proyek. Aktifitas yang dilakukan seperti serah terima proyek atau seremonial penutupan proyek serta mengambil pelajaran penting, yang membantu meningkatkan efektifitas proyek di masa depan.

### 2.5 Algoritma Genetika dalam Penjadwalan

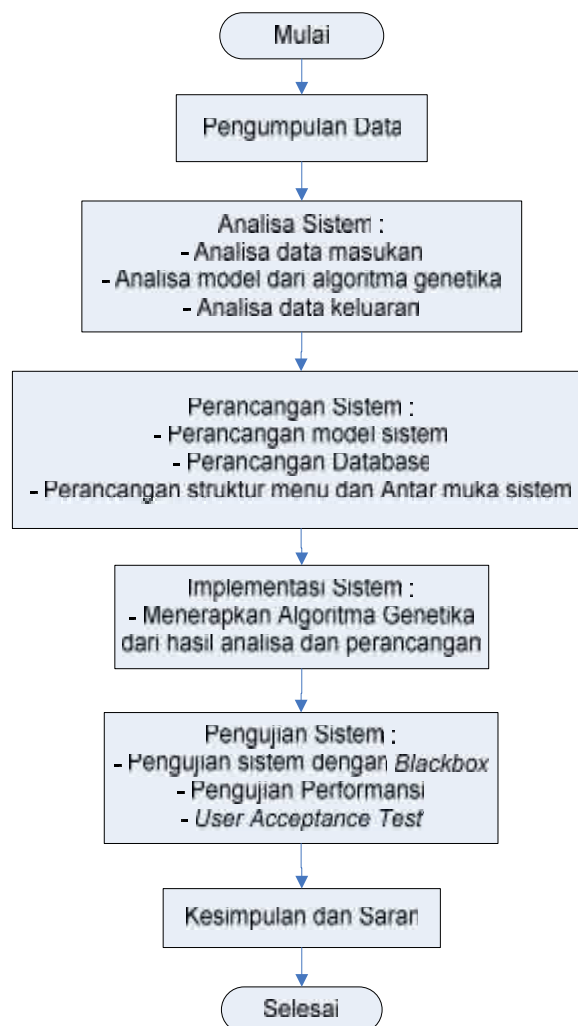
Beberapa unsur penjadwalan yang mendukung pencarian genetika adalah (Susetyo,2004 dikutip dari Yendrika Putra,2009):

- a. Penjadwalan dapat direpresentasikan sebagai kromosom.
- b. Fungsi *fitness* dapat didefinisikan agar algoritma dapat mengetahui seberapa baik solusi tersebut. Hal penting yang perlu diketahui bahwa fungsi *fitness* penjadwalan dapat membedakan jadwal yang paling optimal dengan yang buruk.
- c. Solusi berdasarkan algoritma genetika dapat mengakomodasi *constraint* dalam permasalahan penjadwalan.
- d. Ruang pencarian permasalahan penjadwalan biasanya sangat luas dan multi-modal. Pencarian genetika sangat cocok untuk permasalahan ini.

### BAB III

## METODOLOGI PENELITIAN

Metodologi penelitian merupakan rangkaian tahapan penelitian yang tersusun secara sistematis. Tujuan dari metodologi penelitian adalah agar pelaksanaan penelitian mendapatkan hasil yang sesuai dengan tujuan penelitian. Adapun rangkaian tahapan yang akan dilakukan dalam penelitian ini adalah sebagai berikut :



Gambar 3.1 Metodologi Penelitian.

### 3.1 Pengumpulan Data

Pengumpulan data merupakan metode yang difungsikan untuk memperoleh informasi-informasi atau data-data terhadap kasus yang menjadi permasalahan dalam laporan tugas akhir ini. Hal yang dibutuhkan oleh penulis adalah informasi-informasi mengenai penjadwalan proyek dan metode yang digunakan dalam penelitian kasus ini, yaitu algoritma genetika. Ada tiga pendekatan yang penulis lakukan untuk memperoleh informasi-informasi atau pengumpulan data ini diantaranya adalah:

a. Wawancara (*Interview*)

Wawancara yaitu suatu model data dengan mengajukan pertanyaan-pertanyaan atau tanya jawab secara langsung kepada *project manager* dan karyawan CV. Kambang Putra. Wawancara pada penelitian ini dilakukan untuk mengetahui permasalahan-permasalahan atau kendala-kendala dalam penjadwalan proyek.

b. Studi Pustaka (*Library Research*)

Studi pustaka merupakan metode yang dilakukan untuk menemukan dan mengumpulkan data atau informasi kasus dari referensi-referensi terkait. Referensi-referensi ini dapat berupa buku-buku tentang metode penjadwalan, jurnal-jurnal atau tulisan penelitian penjadwalan menggunakan algoritma genetika, atau artikel-artikel yang membahas kasus yang sama dengan kasus dalam laporan ini.

c. Observasi

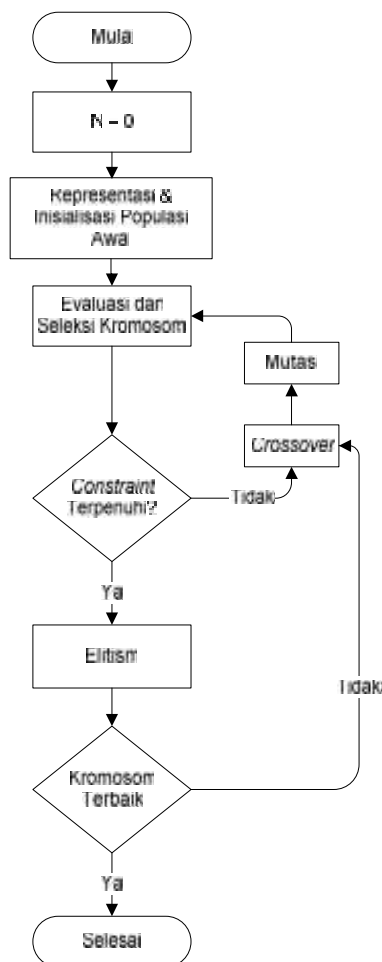
Metode observasi atau pengamatan merupakan salah satu metode pengumpulan data / fakta yang cukup efektif. Observasi merupakan pengamatan langsung yaitu suatu kegiatan yang bertujuan untuk memperoleh informasi yang diperlukan dengan cara melakukan pengamatan dan pencatatan dengan peninjauan langsung ke CV. Kambang Putra.

### 3.2 Analisa Sistem

Tahap ini merupakan tahap analisa terhadap data-data yang telah berhasil dikumpulkan, terdiri dari analisa data masukan, analisa model dari algoritma genetika, dan analisa data keluaran.

1. Analisa data masukan (*input*), bertujuan untuk mengidentifikasi masukan (*input*) yang dibutuhkan oleh sistem.
2. Analisa model dari algoritma genetika untuk penjadwalan proyek.

Gambar 3.2 dibawah ini merupakan *flowchart* penjadwalan proyek menggunakan algoritma genetika.



Gambar 3.2 *Flowchart* Penjadwalan Proyek menggunakan Algoritma Genetika.

Berdasarkan Gambar 3.2 diatas, langkah-langkah penjadwalan proyek menggunakan algoritma genetika dapat dijelaskan sebagai berikut :

- a. Representasi dan inisialisasi populasi awal, yaitu melakukan inisialisasi solusi yang mungkin kedalam sejumlah kromosom. Sebuah kromosom terdiri dari kumpulan gen. Pada penelitian ini, gen berisi informasi saat pelaksanaan suatu kegiatan atau pekerjaan proyek.
  - b. Evaluasi fungsi *fitness*, yaitu menghitung nilai *fitness* tiap-tiap kromosom. Nilai yang dihasilkan tersebut merepresentasikan seberapa banyak jumlah persyaratan atau aturan penjadwalan yang dilanggar.
  - c. Setelah didapatkan nilai *fitness* tiap-tiap kromosom, dilakukan proses seleksi. Pada penelitian ini menggunakan seleksi roda *roulette (Roulette Wheel Selection)*.
  - d. Tentukan probabilitas *crossover*  $P(c)$  lalu bandingkan dengan bilangan acak  $R(i)$  antara 0-1 untuk setiap kromosom yang terdapat pelanggaran. Jika  $R(i)$  kurang dari  $P(c)$ , maka pilih kromosom tersebut sebagai calon induk yang akan dikenai operasi persilangan.
  - e. Kromosom yang masih terdapat pelanggaran akan melakukan proses selanjutnya yaitu mutasi. Sedangkan kromosom yang tidak terdapat pelanggaran akan disimpan untuk dibandingkan dengan kromosom hasil mutasi. *Looping* akan terus dilakukan sampai menghasilkan kromosom terbaik yang tidak terdapat pelanggaran lagi terhadap aturan penjadwalan proyek hingga akhirnya dihasilkan jadwal yang optimal dari kromosom-kromosom tersebut.
3. Analisa data keluaran, merupakan penjelasan dari *output* yang dihasilkan oleh sistem.

### 3.3 Perancangan Sistem

Perancangan berarti metode yang khusus digunakan untuk merancang hal-hal yang telah dianalisa dengan tujuan untuk memberikan kemudahan dan menyederhanakan suatu proses atau jalannya aliran data, perancangan terhadap model sistem, dan merancang bangun sistem ini. Perancangan ini meliputi :

- a. Perancangan model sistem, merupakan gambaran atau rencana alur proses dari sistem yang akan dibangun. Perancangan ini terdiri dari perancangan *flowchart system*, *context diagram*, *data flow diagram* (DFD).
- b. Perancangan *database*, merupakan tahapan untuk memetakan model konseptual ke model basis data yang akan dipakai. Perancangan ini terdiri dari *entity relationship diagram* (ERD) dan kamus data.
- c. Perancangan struktur menu, digunakan untuk memberikan gambaran terhadap menu-menu atau fitur pada sistem yang akan dibangun.
- d. Perancangan antar muka (*interface*) untuk mempermudah komunikasi antara pengguna (*user*) dengan sistem. Dalam perancangan *interface* ini ada hal yang harus diperhatikan yaitu bagaimana menciptakan tampilan yang baik dan mudah dimengerti oleh pengguna.

### 3.4 Implementasi

Pada tahap ini dilakukan penyusunan perangkat lunak sistem (*coding*) sesuai dengan hasil perancangan sistem yang telah dibuat. Untuk mengimplementasikan sistem ini maka dibutuhkan perangkat pendukung, perangkat tersebut berupa perangkat keras dan perangkat lunak.

#### 1. Perangkat keras

Processor : *Intel(R) Core(TM)2 Duo CPU T5750 @ 2.00GHz*  
 Memori (RAM) : 1.00 GB

#### 2. Perangkat Lunak

Sistem operasi : *Windows 7 Ultimate 32-bit OS*  
 Bahasa pemrograman : PHP 5  
 Database : MySQL

### 3.5 Pengujian

Selanjutnya dilakukan pengujian (*testing*) terhadap perangkat lunak yang telah dibangun agar dapat diketahui hasilnya. Jika terdapat *error*, maka pengujian

akan diulangi untuk dilakukan pengecekan ulang. Metode pengujian yang digunakan yaitu:

- a. Pengujian *blackbox* untuk pengujian tingkah laku sistem yang telah dirancang.
- b. Pengujian performansi untuk mengetahui apakah sistem penjadwalan proyek menggunakan algoritma genetika berhasil menghasilkan jadwal yang memenuhi semua *constraint*.
- c. *User Acceptance Test*, merupakan pengujian yang dilakukan dengan meminta persetujuan dari *user* terhadap *output* yang dihasilkan oleh sistem penjadwalan proyek menggunakan algoritma genetika.

### **3.6 Kesimpulan dan Saran**

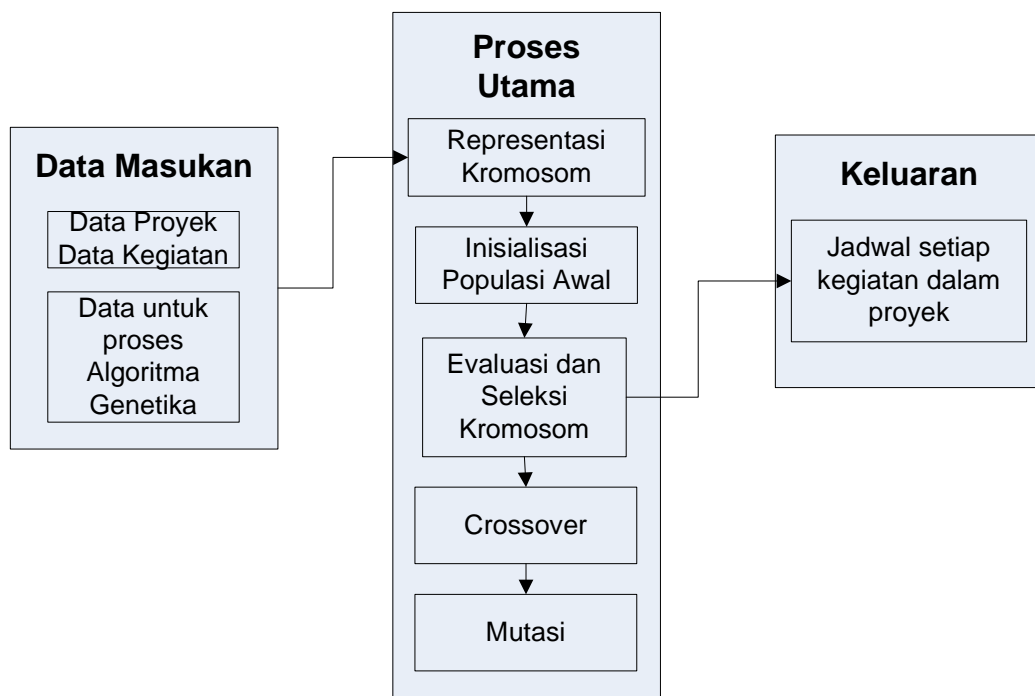
Tahapan kesimpulan dan saran merupakan akhir dari penelitian tugas akhir ini. Tahapan ini berisi tentang kesimpulan dari hasil-hasil penelitian dan pengujian yang telah dilakukan pada penelitian tugas akhir ini, yaitu membangun sistem penjadwalan proyek menggunakan algoritma genetika serta berisi saran-saran membangun yang dapat dijadikan bahan penelitian ulang untuk meneliti dan merancang sistem penjadwalan yang lebih baik.

## BAB IV

### ANALISA DAN PERANCANGAN

#### 4.1 Gambaran Umum Sistem

Implementasi untuk penjadwalan proyek ini berupa sebuah sistem penjadwalan yang menerapkan model dari algoritma genetika. Untuk menyelesaikan permasalahan penjadwalan proyek ini diasumsikan bahwa setiap kegiatan dalam proyek mempunyai durasi dan biaya pelaksanaan kegiatan yang tetap artinya sesuai dengan analisa teknis yang dibuat pada saat perencanaan proyek. Data masukan untuk melakukan penjadwalan adalah data kegiatan yang harus dimasukkan sesuai urutan kerja, dimana sebuah kegiatan tidak boleh diinputkan sebelum input kegiatan predesesornya, yang selanjutnya akan diproses dengan perhitungan menggunakan algoritma genetika. Untuk lebih jelasnya gambaran umum sistem dapat dilihat pada Gambar 4.1 berikut.



Gambar 4.1 Gambaran Umum Sistem.



Berdasarkan Gambar 4.1 dapat diketahui bahwa rancang bangun dari sistem penjadwalan ini memiliki beberapa proses berupa data masukan, proses utama dan keluaran yang dapat dijelaskan sebagai berikut:

1. Data masukan yang dibutuhkan oleh sistem berupa :
  - a. Data proyek, berisi informasi proyek meliputi nama proyek, deskripsi proyek, lokasi proyek, tanggal mulai proyek, dan durasi proyek.
  - b. Data kegiatan, berisi informasi kegiatan dalam proyek meliputi nama kegiatan, durasi kegiatan dan kegiatan pendahulu.
  - c. Data masukan yang diperlukan untuk proses algoritma genetika meliputi data populasi, probabilitas *crossover*, probabilitas mutasi, dan jumlah generasi.
2. Proses Utama, menyatakan proses-proses utama yang terdapat pada rancang bangun sistem penjadwalan menggunakan algoritma genetika, yaitu dimulai dari representasi dan inisialisasi populasi awal, evaluasi nilai *fitness* dan seleksi roda roulette (*Roulette Wheel Selection*), pindah silang (*crossover*) dilakukan dengan penyilangan satu titik (*One-point Crossover*) berdasarkan probabilitas *crossover* yang telah ditentukan dan mutasi berdasarkan probabilitas mutasi yang telah ditentukan. Setelah melalui beberapa generasi, maka algoritma ini akan menghasilkan kromosom terbaik yang merupakan solusi optimal.
3. Keluaran merupakan hasil dari proses-proses utama yang terjadi pada sistem. *Output* dari sistem berupa jadwal setiap kegiatan dalam sebuah proyek yang ditampilkan dalam bentuk kalender kerja agar lebih jelas serta mudah dimengerti.

#### **4.1.1 Analisa Data Masukan (*Input*)**

Dari gambaran umum sistem penjadwalan pada Gambar 4.1 telah dijelaskan bahwa dalam prosesnya sistem penjadwalan ini menggunakan data sebuah proyek sebagai data masukan. Didalam data proyek terdapat data kegiatan yaitu rincian pekerjaan-pekerjaan atau kegiatan-kegiatan dalam sebuah proyek, Data masukan kegiatan terdiri dari nama kegiatan, durasi kegiatan, dan kegiatan

pendahulu yang harus diinputkan berurutan. Waktu yang digunakan dalam sistem ini adalah minggu berdasarkan perhitungan yang biasa digunakan di CV. Kambang Putra. Data kegiatan tersebut sangat berguna dalam proses algoritma genetika sebagai aturan untuk menghasilkan solusi yang optimal. Data masukan yang digunakan dalam sistem penjadwalan proyek ini didapatkan dari analisa teknis yang sudah disetujui.

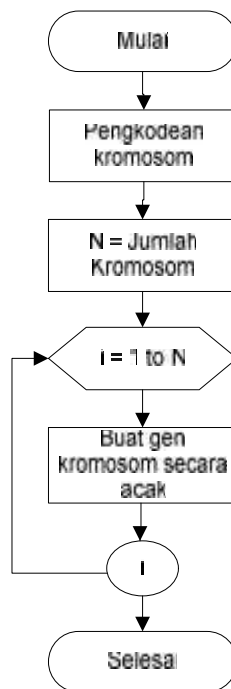
#### **4.1.2 Analisa Algoritma Genetika dalam Penjadwalan Proyek**

Aliran proses penjadwalan proyek dengan menggunakan algoritma genetika mempunyai beberapa prosedur utama yang digambarkan pada Gambar 3.2.

Berikut penjelasan secara detil tentang langkah-langkah algoritma genetika yang akan dipakai untuk penyelesaian persoalan.

##### **1. Representasi dan Inisialisasi Populasi Awal**

Menentukan populasi awal adalah proses membangkitkan sejumlah kromosom secara acak (*random*). Pada penelitian ini, sebuah kromosom menyatakan sebuah solusi penjadwalan. Kromosom terdiri dari kumpulan gen yang berisi informasi saat pelaksanaan suatu kegiatan atau pekerjaan proyek. Panjang kromosom adalah  $n$ , dengan  $n$  adalah banyaknya kegiatan atau pekerjaan. Jadi, pada sistem ini satu gen mewakili waktu mulai pelaksanaan satu kegiatan atau pekerjaan. Teknik pengkodean kromosom yang dipakai yaitu pengkodean bilangan bulat. Pemilihan ini didasarkan pada kromosom yang akan dikodekan sebagai solusi dari waktu pelaksanaan (minggu ke-), sehingga teknik pengkodean ini dinilai cocok untuk dipilih. Nilai setiap gen adalah bilangan bulat yang diperoleh secara acak (*random*) dengan interval yang telah ditentukan (durasi proyek). Secara umum, *flowchart* pembentukan populasi awal dapat dilihat pada gambar dibawah ini :



Gambar 4.2 *Flowchart* Pembentukan Populasi Awal.

Berdasarkan gambar diatas dapat dijelaskan bahwa :

- a. Proses mencari solusi penjadwalan dimulai dari membentuk populasi awal yaitu dengan mengkodekan solusi permasalahan kedalam bentuk kromosom. Ukuran populasi atau jumlah kromosom tergantung pada masalah yang akan diselesaikan.
  - b. Setelah ukuran populasi ditentukan, kemudian dilakukan pembangkitan populasi awal dengan cara melakukan inisialisasi (i) solusi yang mungkin kedalam sejumlah kromosom.
  - c. Setiap kromosom berisi gen yang mewakili waktu mulai pelaksanaan satu kegiatan atau pekerjaan.
  - d. Selesai.
2. Evaluasi Fungsi Fitness

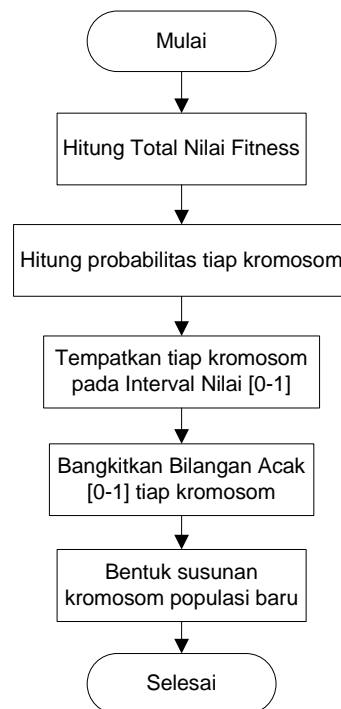
Suatu kromosom dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran performansinya. Didalam evolusi alam, individu yang bernilai *fitness* tinggi yang akan bertahan hidup. Sedangkan individu yang bernilai *fitness* rendah akan mati.

Fungsi yang digunakan untuk mengukur nilai kecocokan atau derajat optimalitas suatu kromosom disebut dengan *fitness function*. Nilai yang dihasilkan dari fungsi tersebut menandakan seberapa optimal solusi yang diperoleh. Nilai yang dihasilkan oleh fungsi *fitness* juga merepresentasikan seberapa banyak jumlah persyaratan yang dilanggar, sehingga dalam kasus penjadwalan proyek semakin kecil jumlah pelanggaran yang dihasilkan maka solusi yang dihasilkan akan semakin baik. Untuk setiap pelanggaran yang terjadi akan diberikan nilai 1. Agar tidak terjadi nilai *fitness* yang tak terhingga maka jumlah total semua pelanggaran akan ditambahkan 1.

### 3. Seleksi

Proses seleksi yang dilakukan menggunakan seleksi roda roulette (*Roulette Wheel Selection*). Adapun langkah-langkah proses seleksi ini dapat dilihat pada gambar 4.3 berupa *flowchart* yang dapat dijelaskan sebagai berikut :

- a. Hasil nilai *fitness* yang didapat tiap kromosom, lalu dijumlahkan.
- b. Lalu hitung nilai probabilitas tiap kromosom
- c. Setelah itu, tentukan interval nilai kumulatif [0-1] tiap kromosom
- d. Pilih induk yang menjadi kandidat untuk populasi baru dengan cara : bangkitkan suatu bilangan random [0-1] dan kromosom yang akan terpilih jika bilangan random yang dibangkitkan berada dalam interval kumulatifnya.
- e. Lakukan langkah 4 sebanyak ukuran populasi, sehingga terbentuk susunan populasi baru.
- f. Selesai.



Gambar 4.3 *Flowchart* Proses Seleksi.

#### 4. Pindah Silang (*crossover*)

Secara umum, langkah-langkah proses *crossover* ditunjukkan pada Gambar 4.4 dapat dijelaskan bahwa:

- Pertama, tentukan probabilitas persilangan  $P(c)$ .
- Bilangan acak  $R(i)$  antara 0-1 yang dibangkitkan untuk setiap kromosom pada proses seleksi dibandingkan dengan  $P(c)$ .
- Jika  $R(i)$  kurang dari  $P(c)$ , maka pilih kromosom tersebut sebagai calon induk yang akan dikenai operasi persilangan.
- Tentukan satu titik potong secara acak tergantung panjang kromosom dan lakukan operasi persilangan pada semua pasangan kromosom secara berurutan, yaitu kromosom 1 akan disilangkan dengan kromosom 2, kromosom 2 dengan kromosom 3, begitu seterusnya.
- Selesai.



Gambar 4.4 *Flowchart* Proses Pindah Silang.

Metode *crossover* yang sering digunakan pada algoritma genetika adalah *one-point crossover* atau penyilangan satu titik. Setelah didapatkan hasil *crossover*, kemudian dilakukan evaluasi kembali terhadap kromosom tersebut. Kromosom yang tidak mengandung pelanggaran akan disimpan, sedangkan kromosom yang mengandung pelanggaran akan dilakukan proses mutasi.

## 5. Mutasi

Proses mutasi adalah suatu proses kemungkinan memodifikasi informasi gen-gen pada suatu kromosom. Perubahan ini dapat membuat solusi duplikasi memiliki nilai *fitness* lebih rendah maupun lebih tinggi daripada solusi induknya. Tetapi jika diperoleh solusi dengan nilai *fitness* yang lebih rendah maka bisa jadi pada iterasi berikutnya diperoleh solusi hasil mutasi yang lebih baik nilai *fitness*nya daripada solusi induknya. Secara detail, *flowchart* proses ini ditunjukkan pada Gambar 4.5 dapat dijelaskan bahwa :

- a. Proses mutasi dimulai dengan menentukan probabilitas mutasi.

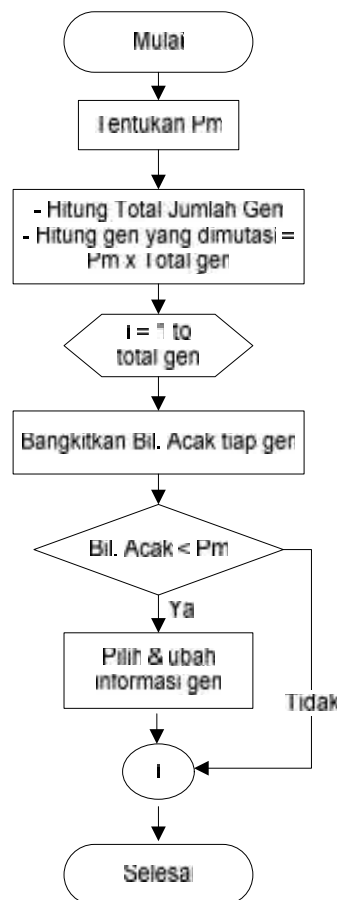
- b. Setelah itu, menghitung total gen dan jumlah mutasi:

Total gen = jumlah gen dalam suatu kromosom \* jumlah kromosom yang ada.....(Persamaan 4.1)

Jumlah Mutasi = Probabilitas mutasi ( $P_m$ ) \* total gen..(Persamaan 4.2)

Proses mutasi dilakukan sebanyak jumlah mutasi yang diperoleh.

- c. Tentukan posisi gen yang akan dimutasi dengan membangkitkan bilangan acak antara 0 sampai 1 tiap gen. Lalu bandingkan dengan probabilitas mutasi.
- d. Jika kecil dari probabilitas mutasi maka pilih dan ubah informasi gen tersebut. Nilai gen yang terkena proses mutasi nilainya akan diganti secara random dengan nilai yang terdapat pada tabel yang mewakili nilai tersebut.
- e. Selesai.



Gambar 4.5 Flowchart Proses Mutasi.

Setelah proses berhenti, kromosom hasil mutasi akan dievaluasi. Apabila masih ditemukan kromosom yang mengandung pelanggaran dari evaluasi yang dilakukan, maka akan kembali dilakukan proses *crossover* dan mutasi sampai sejumlah kromosom yang tersisa habis dan tidak mengandung pelanggaran lagi.

#### 6. *Elitism*

Proses ini adalah untuk membuat salinan (*copy*) individu bernilai *fitness* tertinggi agar tidak hilang selama proses evolusi.

#### 7. Kondisi Selesai

Kondisi selesai yang dapat menghentikan proses algoritma genetika ini adalah jika jumlah generasi atau iterasi maksimum telah tercapai.

Adapun sebagai contoh penyelesaian persoalan, maka digunakan data proyek pembangunan Sistem Pelayanan Air Minum Kawasan Koto Baru Muara Labuah Kabupaten Solok Selatan dengan rincian pekerjaan atau kegiatan dapat dilihat pada tabel di bawah ini berikut penjelasan susunan kegiatannya:

Tabel 4.1 Rincian kegiatan

No	Nama Kegiatan	Kegiatan Pendahulu ( <i>predecessor</i> )	Durasi (Minggu)
1	Pengadaan/Pemasangan Pipa Distribusi		
	1.1 Pekerjaan Pendahuluan	-	1
	1.2 Pengadaan Material	1.1	2
	1.3 Pekerjaan Tanah	1.1	5
	1.4 Pekerjaan Pemasangan Pipa	1.2	4
	1.5 Pekerjaan Lain-Lain	1.2	1
2	Pekerjaan Jembatan Diameter 100 mm- L = 50 m		
	1.6 Pekerjaan Permulaan	1.3,1.4,1.5	1
	1.7 Pekerjaan Pondasi Tiang	1.3,1.4,1.5	1
	1.8 Pekerjaan Tiang	1.6,1.7	1
	1.9 Pekerjaan Tumpuan Kabel	1.6,1.7	2
	1.10 Pekerjaan Kedudukan Pipa	1.8	2
	1.11 Pekerjaan Penggantung	1.8	1
	1.12 Pekerjaan Pipa dan Asesoris	1.9,1.11	1
3	Pekerjaan Jembatan Pipa Diameter 75 mm – L = 25 m		



	1.13 Pembuatan Jembatan Tempel	1.9,1.11	1
	1.14 Pekerjaan Finishing	1.12,1.13	1
Total Waktu (Minggu)			24

Langkah-langkah perhitungan :

### 1. Representasi dan Inisialisasi Populasi Awal

Contoh representasi kromosom suatu populasi dengan ukuran populasi 8 dari kasus pada contoh diatas dapat dilihat pada tabel dibawah ini.

Tabel 4.2 Inisialisasi Populasi Awal.

	X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	X <sub>5</sub>	X <sub>6</sub>	X <sub>7</sub>	X <sub>8</sub>	X <sub>9</sub>	X <sub>10</sub>	X <sub>11</sub>	X <sub>12</sub>	X <sub>13</sub>	X <sub>14</sub>
Kromosom 1	1	2	2	4	5	7	8	9	9	10	10	11	11	12
Kromosom 2	1	2	3	4	5	7	8	8	9	9	10	11	11	12
Kromosom 3	2	3	1	5	6	9	9	10	10	11	11	12	12	10
Kromosom 4	2	1	4	5	8	9	9	10	10	11	11	12	12	13
Kromosom 5	1	2	1	4	7	8	7	9	9	10	10	11	11	12
Kromosom 6	1	2	3	2	1	7	8	8	9	11	11	12	12	13
Kromosom 7	1	2	2	3	6	9	9	10	10	11	11	12	12	13
Kromosom 8	1	1	2	2	5	8	8	9	9	10	10	11	11	12

Dari tabel diatas untuk kromosom 1 terdiri dari kumpulan gen X<sub>1</sub>, X<sub>2</sub>,.....X<sub>14</sub> berisi bilangan bulat positif yang merupakan saat pelaksanaan kegiatan proyek. Pada kromosom 1 dapat dijelaskan bahwa gen X<sub>1</sub> mempunyai nilai 1 menunjukkan bahwa kegiatan 1.1 dilaksanakan pada minggu ke 1, gen X<sub>2</sub> mempunyai nilai 2 menunjukan bahwa kegiatan 1.2 dilaksanakan pada minggu ke 2, dan seterusnya sampai dengan gen X<sub>14</sub> yang mempunyai nilai 12 yang menunjukan bahwa kegiatan 1.14 dilaksanakan pada minggu ke 12. Demikian juga dengan kromosom ke 2 sampai dengan kromosom ke 8.

### 2. Evaluasi Fungsi Fitness

Proses ini dilakukan dengan memperhatikan aturan (*constraint*) yang telah ditetapkan sebelumnya. Setiap kromosom akan diperiksa satu persatu dan dibandingkan dengan kromosom lainnya sesuai dengan *constraint*.

$$\begin{array}{ll}
 \text{Fitness Kromosom 1} = \frac{1}{1+1} = 0.5 & \text{Fitness Kromosom 5} = \frac{1}{1+3} = 0.25 \\
 \text{Fitness Kromosom 2} = \frac{1}{1+3} = 0.25 & \text{Fitness Kromosom 6} = \frac{1}{1+4} = 0.2 \\
 \text{Fitness Kromosom 3} = \frac{1}{1+3} = 0.25 & \text{Fitness Kromosom 7} = \frac{1}{1+1} = 0.5 \\
 \text{Fitness Kromosom 4} = \frac{1}{1+1} = 0.5 & \text{Fitness Kromosom 8} = \frac{1}{1+2} = 0.33
 \end{array}$$

### 3. Seleksi

Pembentukan susunan kromosom pada suatu populasi baru dilakukan dengan menggunakan metode seleksi *roulette-wheel*. Sesuai dengan namanya, metode ini menirukan permainan *roulette-wheel* dimana masing-masing kromosom menempati potongan lingkaran pada *roulette-wheel* secara proporsional sesuai dengan nilai *fitness*nya. Kromosom yang memiliki nilai *fitness* lebih besar menempati potongan lingkaran yang lebih besar dibandingkan dengan kromosom bernilai *fitness* rendah.

Langkah pertama metode ini adalah menghitung total nilai *fitness* seluruh kromosom.

Tabel 4.3 Total Nilai *Fitness*

Kromosom	Nilai <i>Fitness</i>
1	0.5
2	0.25
3	0.25
4	0.5
5	0.25
6	0.2
7	0.5
8	0.33
<b>Total Nilai <i>Fitness</i></b>	<b>2.78</b>

Langkah kedua adalah menghitung probabilitas setiap kromosom dengan cara membagi nilai *fitness* tiap kromosom dengan total nilai *fitness*. Sehingga didapatkan hasil seperti ini :

Tabel 4.4 Probabilitas tiap kromosom

Kromosom	Probabilitas
1	$0.5 / 2.78 = 0.179$
2	$0.25 / 2.78 = 0.089$

3	$0.25 / 2.78 = 0.089$
4	$0.5 / 2.78 = 0.179$
5	$0.25 / 2.78 = 0.089$
6	$0.2 / 2.78 = 0.072$
7	$0.5 / 2.78 = 0.179$
8	$0.33 / 2.78 = 0.119$
<b>Total Probabilitas</b>	<b>1</b>

Langkah ketiga adalah menempatkan masing-masing kromosom pada interval nilai [0-1].

Tabel 4.5 Interval tiap kromosom

Kromosom	Interval Nilai
1	0 – 0.179
2	0.180 – 0.268
3	0.269 – 0.357
4	0.358 – 0.536
5	0.537 – 0.625
6	0.626 – 0.697
7	0.698 – 0.876
8	0.877 – 1

Untuk menentukan susunan populasi baru hasil seleksi maka dibangkitkan bilangan acak (*random*) antara [0-1]. Dimisalkan bilangan yang dibangkitkan adalah [0.25;0.1;0.35;0.5;0.75;0.6;0.9;0.65] maka susunan kromosom populasi baru hasil seleksi adalah :

Tabel 4.6 Populasi baru hasil seleksi

	X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	X <sub>5</sub>	X <sub>6</sub>	X <sub>7</sub>	X <sub>8</sub>	X <sub>9</sub>	X <sub>10</sub>	X <sub>11</sub>	X <sub>12</sub>	X <sub>13</sub>	X <sub>14</sub>
Kromosom 1	1	2	3	4	5	7	8	8	9	9	10	11	11	12
Kromosom 2	1	2	2	4	5	7	8	9	9	10	10	11	11	12
Kromosom 3	2	3	1	5	6	9	9	10	10	11	11	12	12	10
Kromosom 4	2	1	4	5	8	9	9	10	10	11	11	12	12	13
Kromosom 5	1	2	2	3	6	9	9	10	10	11	11	12	12	13
Kromosom 6	1	2	1	4	7	8	7	9	9	10	10	11	11	12
Kromosom 7	1	1	2	2	5	8	8	9	9	10	10	11	11	12
Kromosom 8	1	2	3	2	1	7	8	8	9	11	11	12	12	13

#### 4. Pindah Silang (*Crossover*)

Pindah silang (*CrossOver*) digunakan sebagai metode pemotongan kromosom secara acak (*random*). Metode pindah silang yang paling umum digunakan adalah pindah silang satu titik potong (*one-point crossover*). Suatu titik potong dipilih secara acak (*random*), kemudian bagian pertama dari kromosom induk 1 digabungkan dengan bagian kedua dari kromosom induk 2. Bilangan acak (*random*) yang dibangkitkan untuk menentukan posisi titik potong adalah  $[1-N]$  dimana  $N$  merupakan banyaknya jumlah gen dalam satu kromosom. Untuk contoh ini ditentukan probabilitas *crossover* ( $P_c$ ) yaitu 0.6.

Dimisalkan dari contoh yang ada, nilai untuk kromosom 1, 2, 3, dan 4 kurang dari probabilitas *crossover* ( $P_c$ ) yang telah ditentukan serta bilangan acak (*random*) untuk posisi titik potong adalah pada posisi gen ke-3, maka proses pindah silangnya adalah :

Tabel 4.7 Kromosom induk

	X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	X <sub>5</sub>	X <sub>6</sub>	X <sub>7</sub>	X <sub>8</sub>	X <sub>9</sub>	X <sub>10</sub>	X <sub>11</sub>	X <sub>12</sub>	X <sub>13</sub>	X <sub>14</sub>
Kromosom 1	1	2	3	4	5	7	8	8	9	9	10	11	11	12
Kromosom 2	<b>1</b>	<b>2</b>	<b>2</b>	<b>4</b>	<b>5</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>9</b>	<b>10</b>	<b>10</b>	<b>11</b>	<b>11</b>	<b>12</b>
Kromosom 3	2	3	1	5	6	9	9	10	10	11	11	12	12	10
Kromosom 4	<b>2</b>	<b>1</b>	<b>4</b>	<b>5</b>	<b>8</b>	<b>9</b>	<b>9</b>	<b>10</b>	<b>10</b>	<b>11</b>	<b>11</b>	<b>12</b>	<b>12</b>	<b>13</b>

Tabel 4.8 Kromosom hasil proses *crossover*

	X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	X <sub>5</sub>	X <sub>6</sub>	X <sub>7</sub>	X <sub>8</sub>	X <sub>9</sub>	X <sub>10</sub>	X <sub>11</sub>	X <sub>12</sub>	X <sub>13</sub>	X <sub>14</sub>
Kromosom 1	1	2	<b>2</b>	<b>4</b>	<b>5</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>9</b>	<b>10</b>	<b>10</b>	<b>11</b>	<b>11</b>	<b>12</b>
Kromosom 2	<b>1</b>	<b>2</b>	3	4	5	7	8	8	9	9	10	11	11	12
Kromosom 3	2	3	<b>4</b>	<b>5</b>	<b>8</b>	<b>9</b>	<b>9</b>	<b>10</b>	<b>10</b>	<b>11</b>	<b>11</b>	<b>12</b>	<b>12</b>	<b>13</b>
Kromosom 4	<b>2</b>	<b>1</b>	1	5	6	9	9	10	10	11	11	12	12	10

$$Fitness \text{ Kromosom 1} = \frac{1}{1+1} = 0.5$$

$$Fitness \text{ Kromosom 2} = \frac{1}{1+3} = 0.25$$

$$Fitness \text{ Kromosom 3} = \frac{1}{1+0} = 1$$

$$Fitness \text{ Kromosom 4} = \frac{1}{1+4} = 0.2$$

Dari hasil *crossover* diatas, kromosom yang masih memiliki pelanggaran yaitu kromosom 1, 2, dan 4 akan menjalani proses selanjutnya yaitu mutasi.

Sedangkan kromosom yang tidak memiliki pelanggaran yaitu kromosom 3 akan disimpan sebagai kromosom yang baik yang nantinya akan digunakan untuk dibandingkan kromosom hasil mutasi.

### 5. Mutasi

Untuk mendapatkan posisi gen yang akan dimutasi maka perlu dihitung jumlah total gen dalam satu populasi yaitu **Total gen = Jumlah gen dalam satu kromosom x Jumlah kromosom yang ada**. Berdasarkan contoh yang ada maka total gen adalah  $= 14 \times 3 = 42$ . Probabilitas mutasi umumnya diset antara [0-1], misalnya 0,1 maka diharapkan mutasi yang terjadi adalah  $: 0,1 \times 42 = 4,2 = 4$  gen yang akan mengalami mutasi. Pemilihan posisi gen yang akan dimutasi dilakukan secara acak. Diasumsikan gen yang mendapatkan bilangan dibawah probabilitas mutasi adalah gen ke 5, 6, 17, dan 25. Nilai gen yang terkena proses mutasi nilainya akan diganti secara *random*. Setelah proses berhenti, kromosom hasil mutasi akan dievaluasi.

Tabel 4.9 Kromosom hasil proses mutasi

	X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	X <sub>5</sub>	X <sub>6</sub>	X <sub>7</sub>	X <sub>8</sub>	X <sub>9</sub>	X <sub>10</sub>	X <sub>11</sub>	X <sub>12</sub>	X <sub>13</sub>	X <sub>14</sub>
Kromosom 1	1	2	2	4	<b>4</b>	<b>8</b>	8	9	9	10	10	11	11	12
Kromosom 2	1	2	<b>1</b>	4	5	7	8	8	9	9	<b>6</b>	11	11	12
Kromosom 4	2	1	1	5	6	9	9	10	10	11	11	12	12	10

$$\begin{aligned}
 \text{Fitness Kromosom 1} &= \frac{1}{1+0} = 1 \\
 \text{Fitness Kromosom 2} &= \frac{1}{1+4} = 0.2 \\
 \text{Fitness Kromosom 4} &= \frac{1}{1+4} = 0.2
 \end{aligned}$$

Hasil dari perhitungan diatas kromosom 1 dan 3 memiliki nilai fitness terbaik karena tidak terdapat pelanggaran terhadap *constraint* yang telah ditetapkan. Karena kegiatan pertama pada kromosom 3 bisa dilaksanakan pada hari pertama, maka kromosom 1 merupakan hasil yang terbaik dan solusi yang paling optimal.

Tabel 4.10 Kromosom terbaik

	X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	X <sub>5</sub>	X <sub>6</sub>	X <sub>7</sub>	X <sub>8</sub>	X <sub>9</sub>	X <sub>10</sub>	X <sub>11</sub>	X <sub>12</sub>	X <sub>13</sub>	X <sub>14</sub>
Kromosom 1	1	2	2	4	4	8	8	9	9	10	10	11	11	12

Pada kromosom 1 dapat dijelaskan bahwa gen  $X_1$  mempunyai nilai 1 menunjukkan bahwa kegiatan 1.1 dilaksanakan pada minggu ke 1, gen  $X_2$  mempunyai nilai 2 menunjukkan bahwa kegiatan 1.2 dilaksanakan pada minggu ke 2, dan seterusnya sampai dengan gen  $X_{14}$  yang mempunyai nilai 12 yang menunjukkan bahwa kegiatan 1.14 dilaksanakan pada minggu ke 12.

#### 4.1.3 Analisa Data Keluaran (*Output*)

*Output* dari sistem penjadwalan proyek ini berupa jadwal setiap kegiatan. Data keluaran juga ditampilkan dalam bentuk kalender kerja agar lebih jelas serta mudah dimengerti sebagai acuan dalam pengerjaan proyek.

#### 4.1.4 Analisa Kebutuhan Fungsi

Sistem penjadwalan proyek membutuhkan beberapa fungsi agar dapat digunakan sebagaimana mestinya oleh *user*, dan memberikan hasil yang optimal. Fungsi yang dibutuhkan *user* adalah sebagai berikut :

- Fungsi *input* data, terdiri dari data proyek, data kegiatan, data untuk proses algoritma genetika, dan data kamus.
- Fungsi proses, terdiri dari proses penjadwalan proyek.
- Fungsi pencetakan laporan, digunakan untuk menampilkan dan mencetak laporan berupa kalender kerja proyek yang dihasilkan oleh sistem.

Sedangkan fungsi-fungsi yang dibutuhkan oleh sistem adalah sebagai berikut :

1. Fungsi Inisialisasi populasi awal

Fungsi ini digunakan untuk penginputan dan inisialisasi populasi awal dengan mengacak semua data waktu pelaksanaan kegiatan menjadi kromosom-kromosom.

2. Fungsi evaluasi dan seleksi kromosom

Fungsi ini digunakan untuk menilai bagus tidaknya sebuah kromosom yang nantinya dapat menjadi solusi dari permasalahan.

3. Fungsi *crossover*

Fungsi ini digunakan untuk melakukan *crossover* (penyilangan) antara dua kromosom yang telah dievaluasi sebelumnya.

4. Fungsi mutasi

Fungsi ini digunakan untuk mengubah susunan gen yang terdapat dalam kromosom dengan mengganti nilai gen.

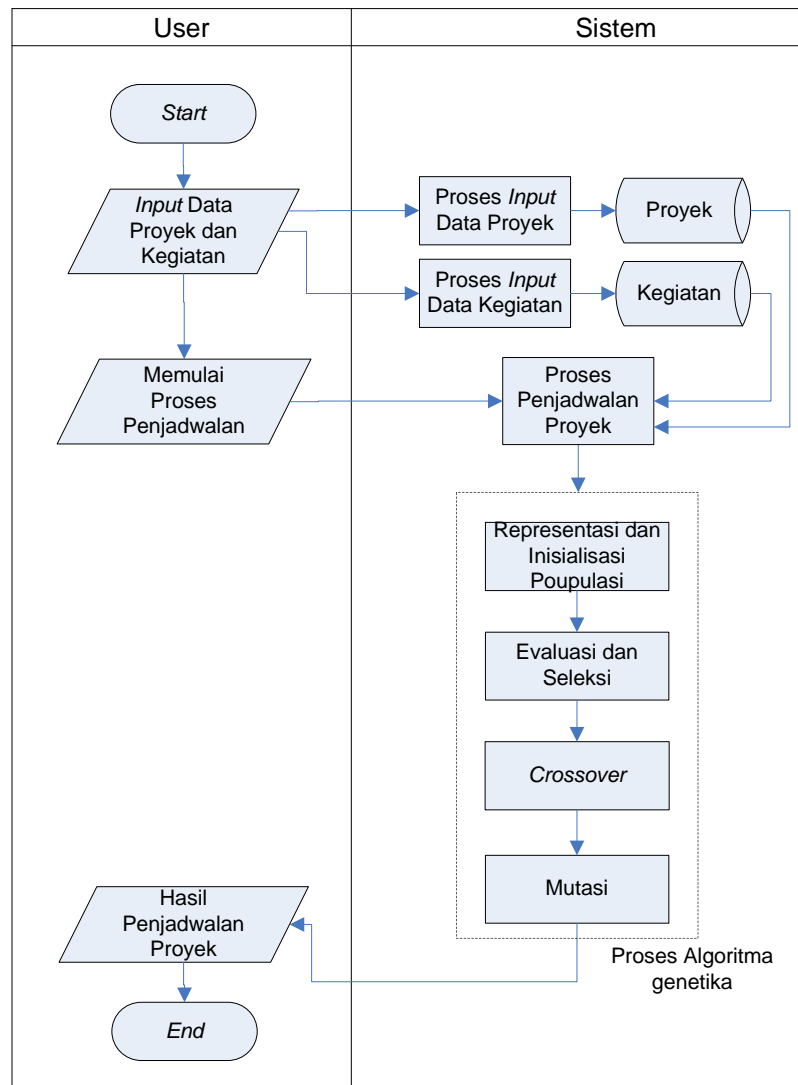
5. Fungsi setting (pengaturan)

Dalam fungsi ini terdapat fasilitas untuk menentukan parameter-parameter dalam algoritma genetika, seperti probabilitas *crossover* dan probabilitas mutasi. Akan tetapi parameter ini juga memiliki nilai default untuk mengantisipasi bila perubahan nilai parameter menghasilkan kinerja yang kurang memuaskan.

## 4.2 Perancangan Sistem

Setelah dilakukan beberapa tahapan dalam analisa sistem, maka dapat dilakukan beberapa perancangan untuk sistem penjadwalan ini. Perancangan-perancangan yang akan dijelaskan dalam laporan ini meliputi perancangan model dalam bentuk *flowchart system*, *context diagram*, *data flow diagram*, *entity relationship diagram*, perancangan basis data, serta perancangan *interface* sistem yang terdiri dari perancangan struktur menu dan perancangan *prototype*.

#### 4.2.1 Flowchart Sistem Yang Dibangun



Gambar 4.6 Flowchart Sistem Yang Dibangun

#### 4.2.2 Context Diagram

Diagram *context* ini merupakan diagram aliran data tingkat atas yang menggambarkan secara umum aliran-aliran data pada sistem. Pada diagram *context* akan terlihat bagaimana arus data yang masuk dan bagaimana arus data keluar dari sistem yang selalu berhubungan dengan entitas luar yang mempengaruhi sistem. Diagram *context* untuk Sistem Penjadwalan Proyek dilihat pada gambar dibawah ini :





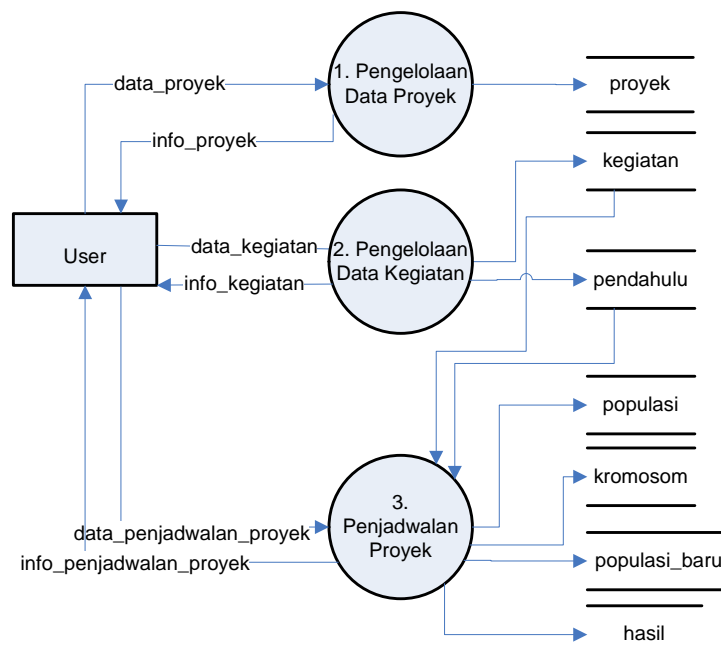
Gambar 4.7 Context Diagram

Entitas luar yang berinteraksi dengan sistem adalah *user*. *User* merupakan pengguna yang dapat memasukkan data kedalam sistem dan mendapatkan hasil dari proses yang dijalankan sistem. *User* adalah manager CV. Kambang Putra.

#### 4.2.3 Data Flow Diagram (DFD)

*Data Flow Diagram* digunakan untuk menggambarkan proses yang ada dalam suatu sistem.

##### a. DFD Level 1 Sistem Penjadwalan Proyek



Gambar 4.8 DFD Level 1 Sistem Penjadwalan Proyek.

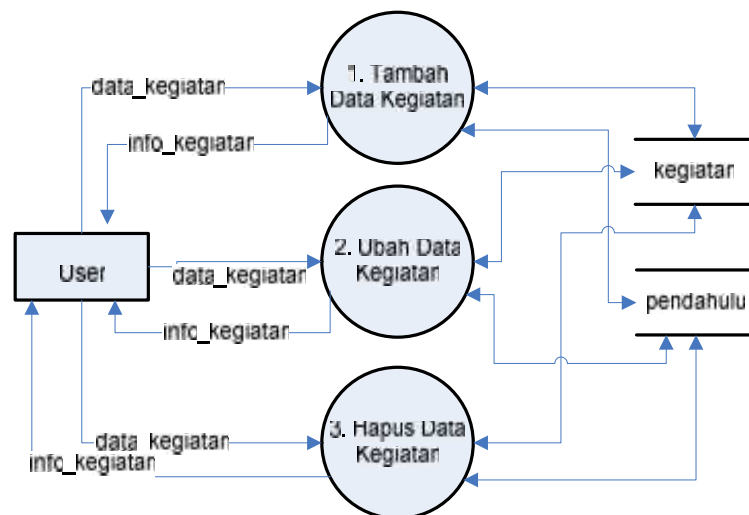
Tabel 4.11 Proses DFD Level 1 Sistem Penjadwalan Proyek

Nama Proses	Deskripsi
Pengelolaan Data Proyek	Berisi proses yang mengelola data proyek ke dalam file penyimpanan.
Pengelolaan Data Kegiatan	Berisi proses yang mengelola data kegiatan ke dalam file – file penyimpanan.
Penjadwalan Proyek	Berisi proses mencari solusi penjadwalan menggunakan algoritma genetika dan menyimpan data ke dalam file-file penyimpanan.

Tabel 4.12 Aliran Data DFD Level 1 Sistem Penjadwalan Proyek

Nama	Deskripsi
data_proyek	Data proyek
data_kegiatan	Data kegiatan dalam proyek
data_penjadwalan_proyek	Data penjadwalan untuk proses algoritma genetika
info_proyek	Informasi data proyek
info_kegiatan	Informasi data kegiatan
Info_penjadwalan_proyek	Informasi solusi penjadwalan yang dihasilkan

## b. DFD Level 2 Pengelolaan Data Kegiatan.



Gambar 4.9 DFD Level 2 Pengelolaan Data Kegiatan.

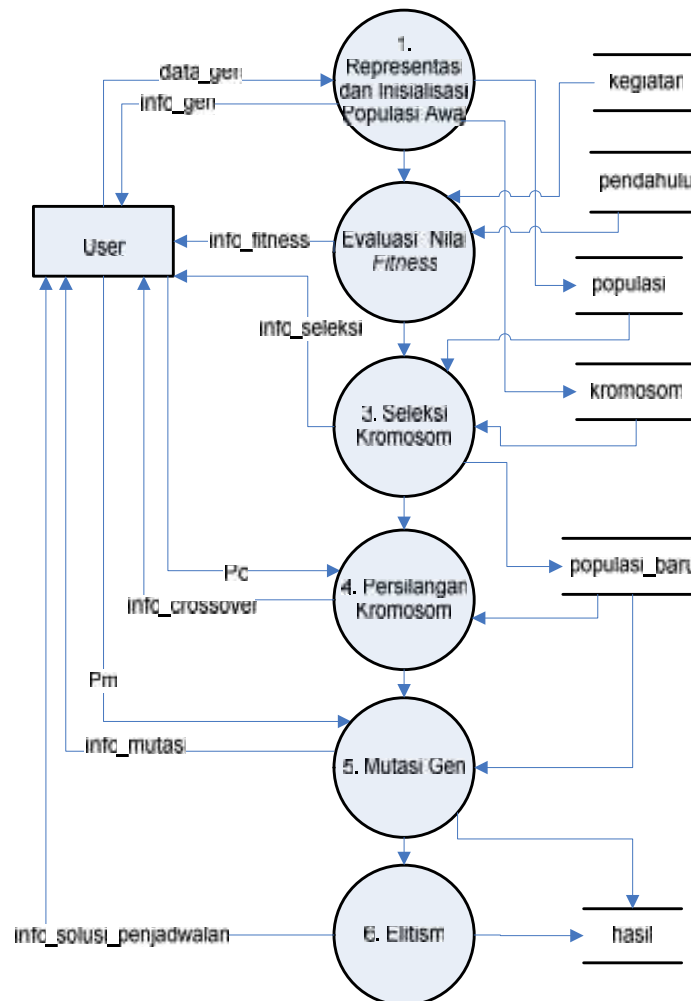
Tabel 4.13 Proses DFD Level 2 Pengelolaan Data Kegiatan

Nama Proses	Deskripsi
Tambah Data Kegiatan	Berisi proses untuk menambah data kegiatan.
Ubah Data Kegiatan	Berisi proses untuk mengubah data kegiatan.
Hapus Data Kegiatan	Berisi proses untuk menghapus data kegiatan.

Tabel 4.14 Aliran Data DFD Level 2 Pengelolaan Data Kegiatan

Nama	Deskripsi
data_kegiatan	Data kegiatan terdiri dari nama kegiatan, durasi kegiatan, dan kegiatan pendahulu.
info_kegiatan	Informasi dari data kegiatan yang telah dimasukkan.

## c. DFD Level 2 Penjadwalan Proyek.



Gambar 4.10 DFD Level 2 Proses Algoritma Genetika.

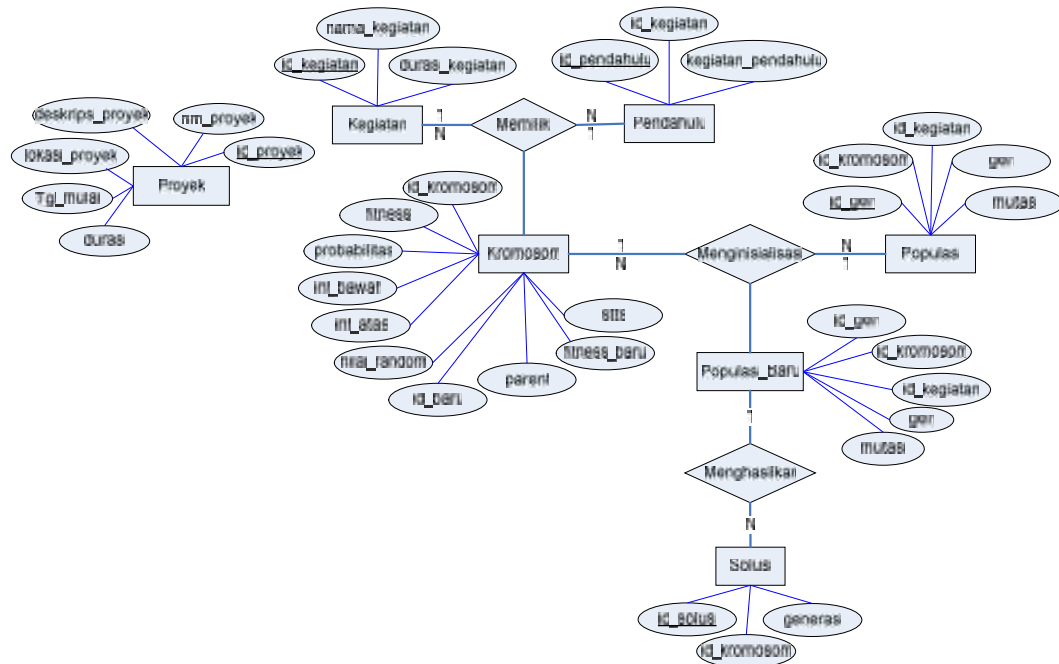
Tabel 4.15 Proses DFD Level 2 Proses Algoritma Genetika

<b>Nama Proses</b>	<b>Deskripsi</b>
Representasi dan Inisialisasi Populasi Awal	Berisi proses pembentukan kromosom dan populasi awal.
Menentukan Nilai <i>Fitness</i>	Berisi proses menentukan nilai <i>fitness</i> setiap kromosom.
Seleksi Kromosom	Berisi proses seleksi kromosom.
Persilangan Kromosom	Berisi proses persilangan kromosom.
Mutasi Gen	Berisi proses mutasi gen.
<i>Elitism</i>	Berisi proses penyimpanan hasil.

Tabel 4.16 Aliran Data DFD Level 2 Proses Algoritma Genetika

<b>Nama</b>	<b>Deskripsi</b>
ukuran_populasi	Data ukuran populasi yang digunakan untuk proses algoritma genetika.
jmlh_generasi	Data jumlah generasi yang digunakan untuk proses algoritma genetika.
Pc	Data probabilitas <i>crossover</i>
Pm	Data probabilitas mutasi
info_populasi	Informasi populasi yang terbentuk
info_fitness	Informasi hasil proses perhitungan nilai fitness.
info_seleksi	Informasi hasil proses seleksi
info_crossover	Informasi hasil proses crossover
info_mutasi	Informasi hasil proses mutasi
Kromosom_akhir	Solusi yang dihasilkan dari proses algoritma genetika.

#### 4.2.4 Entity Relationship Diagram (ERD)



Gambar 4.11 Entity Relationship Diagram.

Tabel 4.17 Keterangan *Entitas* pada ERD

No	Nama	Deskripsi	Atribut	Primary Key
1.	Proyek	Berisi data proyek secara umum yang dibutuhkan sistem.	<ul style="list-style-type: none"> <li>- id_proyek</li> <li>- nm_proyek</li> <li>- deskripsi_proyek</li> <li>- lokasi_proyek</li> <li>- tgl_mulai</li> <li>- durasi</li> </ul>	- <u>id_proyek</u>
2.	Kegiatan	Berisi data kegiatan yang ada dalam sebuah proyek.	<ul style="list-style-type: none"> <li>- id_kegiatan</li> <li>- nama_kegiatan</li> <li>- durasi_kegiatan</li> </ul>	- <u>id_kegiatan</u>
3	Pendahulu	Berisi data kegiatan pendahulu.	<ul style="list-style-type: none"> <li>- id_pendahulu</li> <li>- id_kegiatan</li> <li>- kegiatan_pendahulu</li> </ul>	- <u>id_pendahulu</u>
4	Kromosom	Berisi satu solusi penjadwalan.	<ul style="list-style-type: none"> <li>- id_kromosom</li> <li>- fitness</li> <li>- probabilitas</li> <li>- int_bawah</li> </ul>	

			<ul style="list-style-type: none"> <li>- int_atas</li> <li>- nilai_random</li> <li>- id_baru</li> <li>- parent</li> <li>- fitness_baru</li> <li>- stts</li> </ul>	
5	Populasi	Berisi data kumpulan solusi penjadwalan.	<ul style="list-style-type: none"> <li>- id_gen</li> <li>- id_kromosom</li> <li>- id_kegiatan</li> <li>- gen</li> <li>- mutasi</li> </ul>	- <u>id_gen</u>
6	Populasi Baru	Berisi data kumpulan solusi yang sudah melalui proses algoritma genetika.	<ul style="list-style-type: none"> <li>- id_gen</li> <li>- id_kromosom</li> <li>- id_kegiatan</li> <li>- gen</li> <li>- mutasi</li> </ul>	
7	Solusi	Berisi hasil proses penjadwalan menggunakan algoritma genetika berupa solusi optimal.	<ul style="list-style-type: none"> <li>- id_solusi</li> <li>- id_kromosom</li> <li>- generasi</li> </ul>	- <u>id_solusi</u>

### 4.3 Perancangan Basis Data

Perancangan basis data digunakan untuk membuat detail data yang akan dipersiapkan pada tahap implementasi selanjutnya. *Conceptual* data model digunakan untuk mengetahui tipe-tipe data yang digunakan dalam *database* penjadwalan proyek menggunakan Algoritma Genetika. Berikut *Conceptual* data model dari sistem ini:

Tabel 4.18 *Conceptual* Data Model Tabel Proyek

Field	Type	Null
id_proyek	Integer, 11	No
nama_proyek	Varchar, 200	No
deskripsi_proyek	Text	No
lokasi_proyek	Text	No
tanggal_mulai	Date	No
Durasi	Integer, 11	No

Tabel 4.19 *Conceptual* Data Model Tabel Kegiatan

Field	Type	Null
id_kegiatan	Integer, 11	No
nama_kegiatan	Varchar, 200	No
durasi_kegiatan	Interger, 11	No

Tabel 4.20 *Conceptual* Data Model Tabel Pendahulu

Field	Type	Null
id_pendahulu	Integer, 11	No
id_kegiatan	Integer, 11	No
kegiatan_pendahulu	Varchar, 100	No

Tabel 4.21 *Conceptual* Data Model Tabel Kromosom

Field	Type	Decimals	Null
id_kromosom	Integer, 11	0	No
fitness	Decimal, 10	5	No
probabilitas	Decimal, 10	5	No
int_bawah	Decimal, 10	5	No
int_atas	Decimal, 10	5	No
nilai_random	Decimal, 10	5	No
id_baru	Integer, 10	0	No
parent	Enum	0	No
fitness_baru	Decimal, 10	5	No
stts	Enum	0	No

Tabel 4.22 *Conceptual* Data Model Tabel Populasi

Field	Type	Null
id_gen	Integer, 11	No
id_kromosom	Integer, 10	No
id_kegiatan	Varchar, 10	No
gen	Integer, 11	No
mutasi	Enum	No

Tabel 4.23 *Conceptual* Data Model Tabel Populasi Baru

Field	Type	Null
id_gen	Integer, 11	No
id_kromosom	Integer, 10	No
id_kegiatan	Varchar, 10	No
gen	Integer, 11	No
mutasi	Enum	No

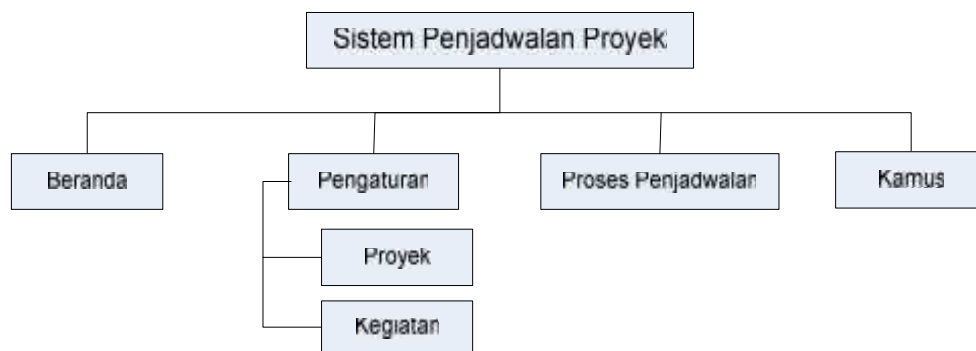
Tabel 4.24 *Conceptual Data Model* Tabel Solusi

Field	Type	Null
id_solusi	Integer, 11	No
id_kromosom	Integer, 11	No
generasi	Integer, 11	No

#### 4.4 Perancangan Antar Muka (*Interface*)

Perancangan antar muka ada beberapa komponen yaitu :

##### 4.4.1 Rancangan Struktur Menu



Gambar 4.12 Struktur Menu.

##### 4.4.2 Rancangan Tampilan Menu Utama

Gambar di bawah ini merupakan tampilan utama dari sistem. Dimana pada menu utama ini *user* dapat mengelola jadwal proyek dengan memilih menu – menu yang terdapat pada Menu Utama ini. Menu – menu pada sistem ini terdiri dari Beranda, Pengaturan, Proses Penjadwalan, dan Kamus. Didalam menu Pengaturan terdapat submenu proyek, submenu kegiatan, dan submenu kamus.



Beranda	Pengaturan	Proses Penjadwalan	Kamus
---------	------------	--------------------	-------

SISTEM PENJADWALAN PROYEK

Gambar 4.13 Rancangan Tampilan Menu Utama.

#### 4.4.3 Rancangan Tampilan Submenu Proyek

**Setting Proyek**

Nama Proyek	<input type="text"/>
Deskripsi Proyek	<input type="text"/>
Lokasi	<input type="text"/>
Tgl. Mulai	<input type="text"/>
Durasi Proyek	<input type="text"/>
<div>Simpan Data    Reset Data</div>	

Gambar 4.14 Rancangan Tampilan Input Data Proyek.

Gambar 4.14 diatas merupakan tampilan untuk menginputkan data-data proyek yang diperlukan dalam membuat penjadwalan proyek.

#### 4.4.4 Rancangan Tampilan Submenu Kegiatan

**Kegiatan**

Nama Kegiatan

Durasi  
 Minggu

Kegiatan Pendahulu

- ☐ Kegiatan 1
- ☐ Kegiatan 2
- ☐ Kegiatan 3
- ☐ Kegiatan 4
- ☐ Kegiatan 5
- ☐ Kegiatan 6
- ☐ Kegiatan 7
- ☐ Kegiatan 8

Gambar 4.15 Rancangan Tampilan Input Data Kegiatan.

Gambar 4.15 diatas merupakan tampilan untuk menginputkan data kegiatan yang diperlukan dalam proses penjadwalan proyek. Pilihan yang terdapat pada inputan kegiatan pendahulu tergantung banyak kegiatan yang diinputkan.

#### 4.4.5 Rancangan Tampilan Proses Penjadwalan Proyek

<div> <div>Proses Inisialisasi</div> <div>Proses Evaluasi</div> <div>Proses Seleksi</div> <div>Crossover</div> <div>Mutasi</div> </div>									
Penjadwalan Proyek Jalan Raya / <span>0.1</span>									
<div> <div>Hasil Mutasi</div> <div>Nilai Fitness</div> </div>									
Hasil Crossover									
No.	Kromosom	X1	X2	X3	X4	X5	X6	X7	X8
1	Kromosom 7	1	2	3	6	7	8	10	15
2	Kromosom 8	2	3	5	6	7	8	10	15
3	Kromosom 7	1	2	3	6	7	8	10	15
4	Kromosom 8	2	3	5	6	7	8	10	15
5	Kromosom 8	2	3	5	6	7	8	10	15
6	Kromosom 7	1	2	3	6	7	8	10	15
Hasil Mutasi									
No.	Kromosom	X1	X2	X3	X4	X5	X6	X7	X8
1	Kromosom 7	1	2	3	4	7	4	10	15
2	Kromosom 8	2	3	2	7	7	8	10	15
3	Kromosom 7	1	2	3	1	7	4	10	15
4	Kromosom 7	1	2	2	7	7	8	10	15
5	Kromosom 8	2	3	2	7	7	8	10	15
6	Kromosom 8	2	3	2	7	7	8	10	15

Gambar 4.16 Rancangan Tampilan Proses Penjadwalan Proyek.

Gambar 4.16 diatas merupakan tampilan untuk menampilkan proses-proses algoritma genetika untuk menghasilkan solusi penjadwalan.

## **BAB V**

### **IMPLEMENTASI DAN PENGUJIAN**

#### **5.1. Tahapan Implementasi**

Tahapan implementasi merupakan tahapan dimana suatu sistem atau perangkat lunak yang telah dianalisa, dirancang dan selanjutnya akan direalisasikan sebagai serangkaian program, lalu diuji kelayakannya. Sehingga akan diketahui bahwa sistem yang dibuat sesuai dengan tujuan yang diinginkan dan dapat dioperasikan sebagaimana mestinya. Berikut ini akan dijelaskan tentang pengimplementasian sistem penjadwalan proyek berdasarkan analisa dan perancangan yang telah dilakukan sebelumnya.

##### **5.1.1. Lingkungan Operasional**

Pada prinsipnya setiap perancangan ataupun pembuatan sebuah sistem memerlukan sarana pendukung yaitu berupa komponen-komponen yang menunjang dalam pembuatan sistem tersebut. Komponen-komponen yang dibutuhkan antara lain *hardware* dan *software*, dimana *hardware* merupakan perangkat keras komputer dalam pengolahan data dan *software* merupakan perangkat lunak berupa sistem untuk mengoperasikan sistem yang akan dibangun.

Lingkungan operasional merupakan lingkungan dimana sistem ini digunakan. Spesifikasi kebutuhan dari lingkungan operasional, yaitu:

#### **1. Perangkat keras**

Processor : *Intel(R) Core(TM)2 Duo CPU T5750 @ 2.00GHz*

Memori (RAM) : 1.00 GB

#### **2. Perangkat Lunak**

Sistem Operasi : *Windows 7 Ultimate 32-bit OS*

Bahasa Pemrograman : PHP 5

### 5.1.2. Implementasi Antar Muka Sistem

Setelah tahap analisa dan perancangan selesai dilakukan, maka dilanjutkan dengan tahap implementasi sistem dari hasil analisa yang telah diperoleh dan mengimplementasikan hasil perancangan *interface* yang telah dibuat. Berikut ini akan dijelaskan mengenai hasil implementasi dari rancang bangun sistem penjadwalan proyek ini, dimana pada sistem penjadwalan ini memiliki empat menu, yaitu menu Beranda, Pengaturan, Penjadwalan Proyek, dan Kamus. Menu Beranda merupakan menu yang akan menampilkan halaman utama sistem. Menu Pengaturan merupakan menu untuk memasukkan data terdiri dari tiga submenu yaitu proyek, kegiatan dan kamus, pada submenu proyek berfungsi untuk menampilkan *form* yang digunakan untuk memasukkan data proyek kedalam *database*, submenu kegiatan berfungsi untuk menampilkan *form* yang digunakan untuk memasukkan data kegiatan kedalam *database*, sedangkan submenu kamus berfungsi untuk menampilkan *form* yang digunakan untuk memasukkan data kamus kedalam *database*. Menu Penjadwalan Proyek berfungsi untuk menampilkan langkah-langkah perhitungan algoritma genetika untuk penjadwalan proyek sampai mendapatkan hasil penjadwalan beserta kalender kerja. Menu Kamus berfungsi untuk pencarian arti kata-kata yang terdapat pada sistem. Berikut implementasi *interface* sistem penjadwalan proyek ini sesuai dengan menu yang ada pada sistem:

#### 1. Implementasi *Interface* Menu Beranda



Gambar 5.1. Hasil Implementasi *Interface* Menu Beranda

*Interface* pada Gambar 5.1 merupakan tampilan yang akan muncul pertama sekali ketika *user* menjalankan sistem ini.

## 2. Implementasi *Interface* Menu Pengaturan

Menu Pengaturan terdiri dari tiga submenu yaitu, proyek, kegiatan dan kamus.

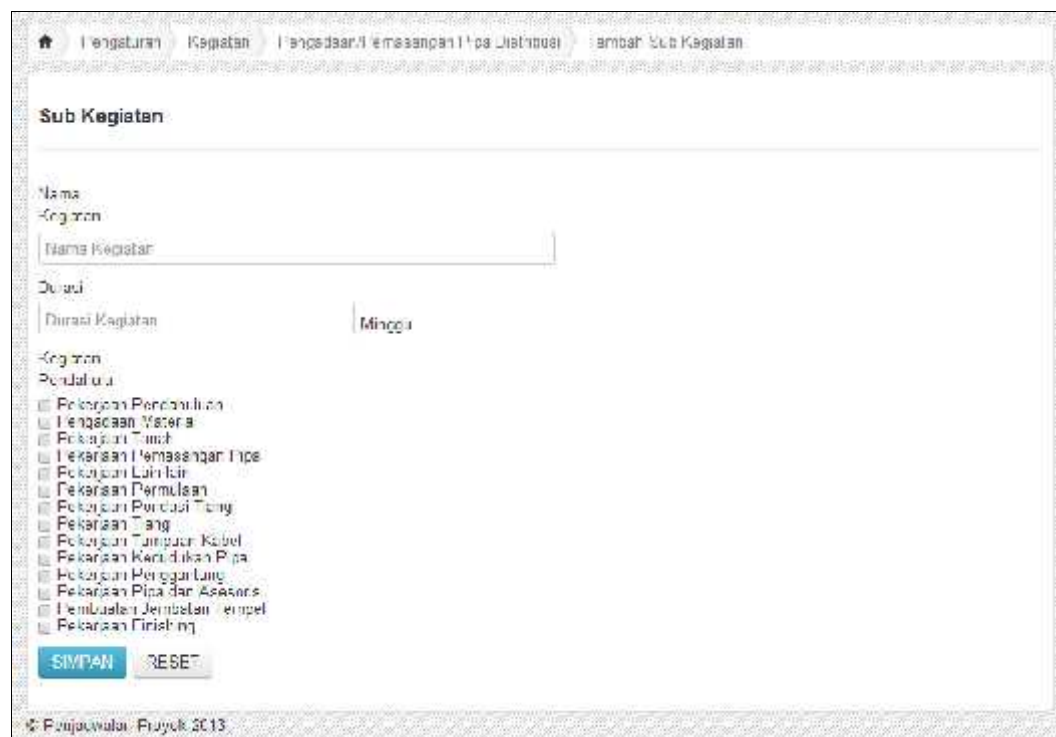
Gambar 5.2. Hasil Implementasi *Interface* Submenu Proyek

*Interface* pada Gambar 5.2 merupakan tampilan dari submenu proyek yang digunakan untuk memasukkan data proyek kedalam database. Pada tampilan submenu proyek ini terdapat dua tombol yaitu simpan data dan reset data. Tombol simpan data digunakan untuk menyimpan data proyek kedalam database dan tombol reset digunakan untuk mengembalikan semua data yang telah diubah ke data semula.

Selanjutnya, pada Gambar 5.3 dapat dilihat *interface* dari submenu kegiatan yang digunakan untuk memasukkan data kegiatan ke dalam database. Setelah mengklik submenu kegiatan akan muncul tampilan data kegiatan yang tersimpan didalam database. Pada tampilan submenu kegiatan ini terdapat tombol tambah kegiatan, tambah sub kegiatan, dan edit kegiatan.



Gambar 5.3. Hasil Implementasi *Interface* Submenu Kegiatan.



Gambar 5.4. Hasil Implementasi *Interface* Input Data Sub Kegiatan.

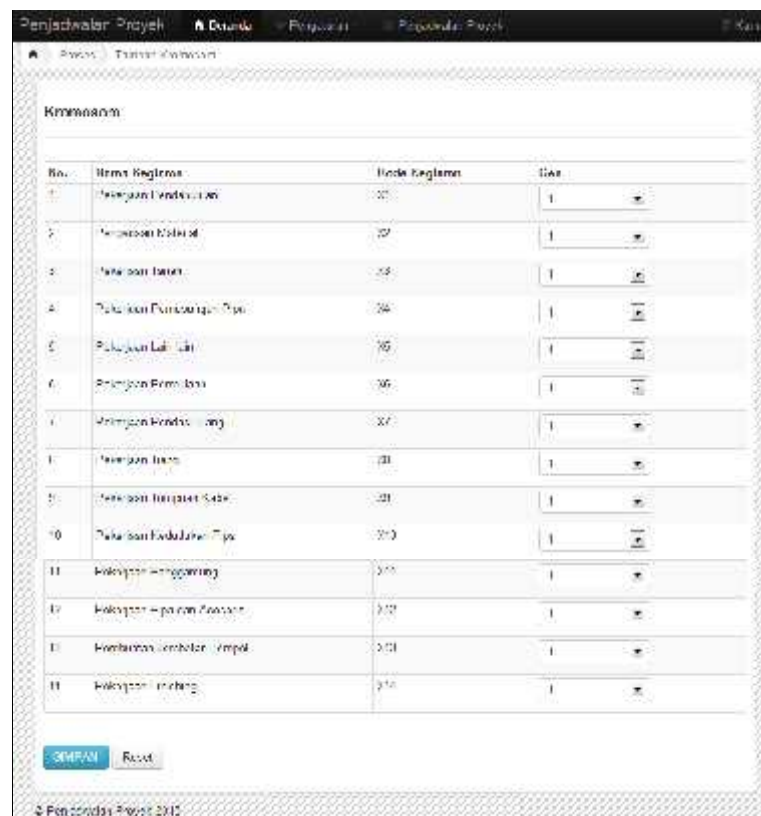
Selanjutnya pada submenu kamus terdapat form inputan data kamus, yaitu berupa keyword dan keterangan.



Gambar 5.5. Hasil Implementasi *Interface Input Data Kamus*.

### 3. Implementasi *Interface Menu Penjadwalan Proyek*

Menu Penjadwalan Proyek berguna untuk menampilkan langkah-langkah mencari hasil penjadwalan proyek menggunakan algoritma genetika yang terdiri dari representasi dan inisialisasi populasi, evaluasi fungsi fitness, seleksi (*roulette-wheel*), pindah silang (*crossover*), dan mutasi.



Gambar 5.6. Hasil Implementasi *Interface Representasi dan Inisialisasi Populasi*.



Gambar 5.6 merupakan tampilan proses pertama dari langkah-langkah penjadwalan proyek, dimana sebuah kromosom menyatakan sebuah solusi penjadwalan. Kromosom berisi kumpulan dari gen yang berisi informasi saat pelaksanaan suatu kegiatan atau pekerjaan proyek. Setelah ukuran populasi ditentukan, kemudian dilakukan pembangkitan populasi awal dengan cara melakukan inisialisasi solusi yang mungkin kedalam sejumlah kromosom yang dapat dilihat pada Gambar 5.7.

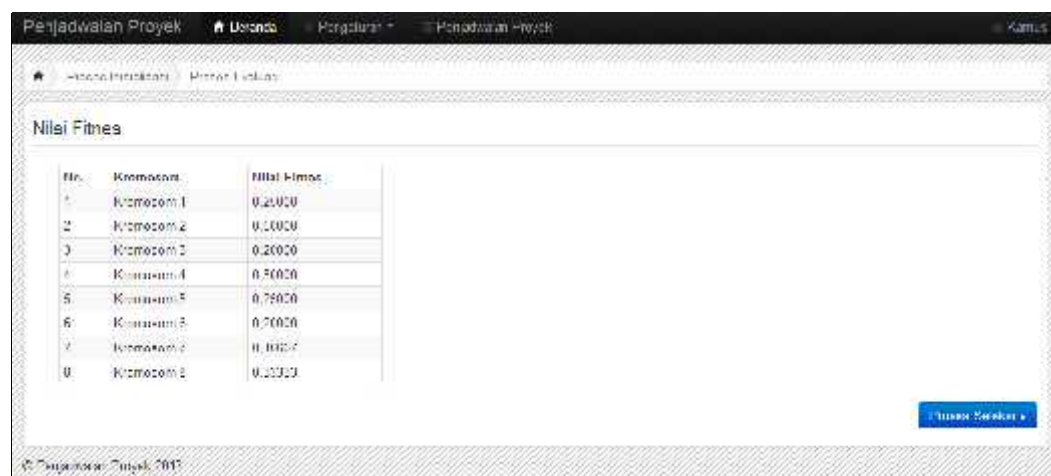


No.	Kromosom	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	Tampilkan
1	Kromosom 1	1	1	3	1	1	1	0	0	0	10	10	11	11	1	
2	Kromosom 2	1	2	3	4	7	8	8	0	0	10	8	11	11	12	
3	Kromosom 3	2	2	1	1	5	5	9	11	10	0	11	12	12	1	
4	Kromosom 4	2	1	4	5	5	5	0	11	10	11	11	12	12	13	
5	Kromosom 5	1	2	1	4	7	8	7	0	0	10	10	11	11	12	
6	Kromosom 6	1	2	2	2	1	1	0	0	0	11	11	12	12	14	
7	Kromosom 7	2	2	4	5	5	5	0	7	7	11	11	12	12	13	
8	Kromosom 8	1	2	2	4	7	0	0	9	9	10	10	11	11	10	

Hitung Nilai Fitness

Gambar 5.7. Hasil Implementasi *Interface* Populasi Awal.

Setelah populasi awal ditentukan, klik tombol hitung nilai fitness untuk proses evaluasi fungsi fitness.

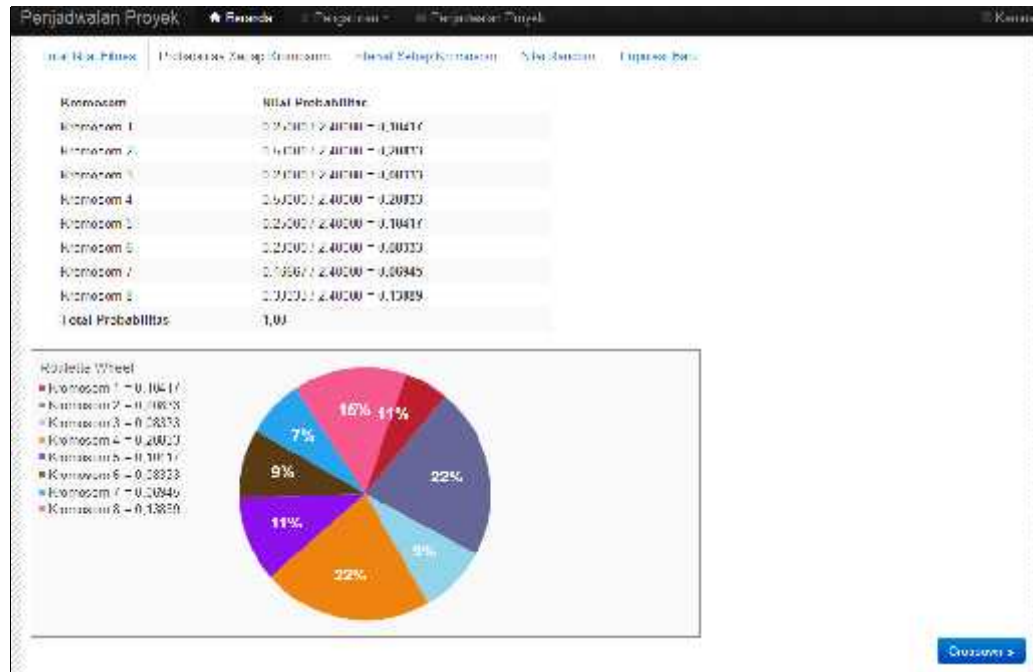


No.	Kromosom	Nilai Fitness
1	Kromosom 1	0.21020
2	Kromosom 2	0.10020
3	Kromosom 3	0.20020
4	Kromosom 4	0.50020
5	Kromosom 5	0.75020
6	Kromosom 6	0.70020
7	Kromosom 7	0.80020
8	Kromosom 8	0.23320

Tampilkan Kembali

Gambar 5.8. Hasil Implementasi *Interface* Proses Evaluasi Fungsi *Fitness*

Setelah mendapatkan hasil nilai *fitness* masing-masing kromosom klik tombol proses seleksi untuk proses selanjutnya.



Gambar 5.9. Hasil Implementasi *Interface* Proses Seleksi

Pada tampilan proses seleksi di atas terdapat langkah-langkah dengan menggunakan metode seleksi *roulette-wheel*, yang terdiri dari menghitung total nilai fitness, probabilitas setiap kromosom, interval setiap kromosom, menentukan nilai random, dan populasi baru. Kemudian klik tombol *crossover* untuk proses selanjutnya.

The screenshot shows a dialog box titled 'Tentukan Nilai pc (Crossover Probability)'. It contains a text area with the following text:

Tips : PC umumnya diseti Mendekati 1, misalnya 0.8 atau 0.9 (Ruyanto, 2005).  
Gunakan (.) titik, sebagai pembatas bilangan desimal

Below the text area is a label 'Crossover Probability / pc' and a text input field with the value '0.8'.

At the bottom of the dialog box are two buttons: 'Cancel' and 'Proses Crossover'.

Gambar 5.10. Hasil Implementasi *Interface* Probabilitas *Crossover*

Pertama tentukan nilai probabilitas *crossover* (pc) terlebih dahulu sebelum memulai proses *crossover*.

Kromosom	Bilangan Acak
Kromosom 1	0.84498
Kromosom 2	0.0264
Kromosom 3	0.00213
Kromosom 4	0.0488
Kromosom 5	0.94642
Kromosom 6	0.93743
Kromosom 7	0.87728
Kromosom 8	0.87582

Gambar 5.11. Hasil Implementasi *Interface* Proses Pindah Silang (*Crossover*)

Dari hasil proses *crossover* diatas, kromosom yang memiliki *error* akan menjalani proses selanjutnya yaitu mutasi. Sedangkan kromosom yang tidak memiliki *error* akan disimpan sebagai kromosom yang baik yang nantinya akan digunakan untuk dibandingkan kromosom hasil mutasi. Untuk proses selanjutnya klik tombol mutasi.

Gambar 5.12. Hasil Implementasi *Interface* Probabilitas Mutasi

Sebelum memulai proses mutasi tentukan terlebih dahulu nilai probabilitas mutasinya. Kemudian klik Tombol Proses Mutasi untuk proses selanjutnya.

Penjadwalan Proyek Urut Penjadwalan Penjadwalan Proyek Kanvas

Home | Jadwal | Urut | Penjadwalan | Penjadwalan Proyek | Kanvas

Proyek Pembangunan Sistem Peningkatan Air Minum (SPAM) / 6.6

Hasil Mutasi Nilai Rms

Hasil Crossover

No.	Kromosom	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14
1	Kromosom 1	1	2	1	4	7	0	5	5	5	12	12	11	11	10
2	Kromosom 5	1	2	1	5	6	9	5	10	10	5	11	12	12	13
3	Kromosom 3	1	2	1	5	7	8	5	5	5	12	12	11	11	10
4	Kromosom 3	1	1	1	5	6	0	5	10	10	5	11	12	12	13
5	Kromosom 7	2	2	4	5	8	0	5	10	10	5	11	12	12	5

Hasil Mutasi

No.	Kromosom	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14
1	Kromosom 1	1	12	1	15	7	8	5	5	5	12	12	11	11	10
2	Kromosom 5	5	5	1	5	6	0	10	10	15	5	11	12	12	5
3	Kromosom 1	1	12	1	15	7	8	5	5	5	12	12	11	11	10
4	Kromosom 5	5	5	1	5	6	0	10	10	15	5	11	12	12	5
5	Kromosom 7	2	2	4	5	8	0	5	10	10	5	11	12	12	5

Re-Generasi

Gambar 5.13. Hasil Implementasi *Interface* Proses Mutasi.

Dari hasil proses mutasi diatas, kromosom yang memiliki *error* akan diproses pada generasi selanjutnya. Sedangkan kromosom yang tidak memiliki *error* akan disimpan sebagai kromosom yang baik yang nantinya akan digunakan untuk dibandingkan dengan hasil yang didapatkan setelah beberapa generasi. Untuk melakukan proses generasi klik tombol re-generasi.

**Tentukan Nilai pc (Crossover Probability)**

Crossover Probability /  $p_c$

Probabilitas Crossover

Probabilitas Mutasi /  $p_{mut}$

Probabilitas Mutasi

Jumlah Generasi

Jumlah Generasi

Cancel Proses Re-Generasi

Gambar 5.14. Hasil Implementasi *Interface* Proses Re-Generasi.

Setelah proses re-generasi selesai, akan keluar kromosom terbaik yang merupakan solusi penjadwalan proyek yang dapat dilihat pada Gambar 5.15.



Penjadwalan Proyek

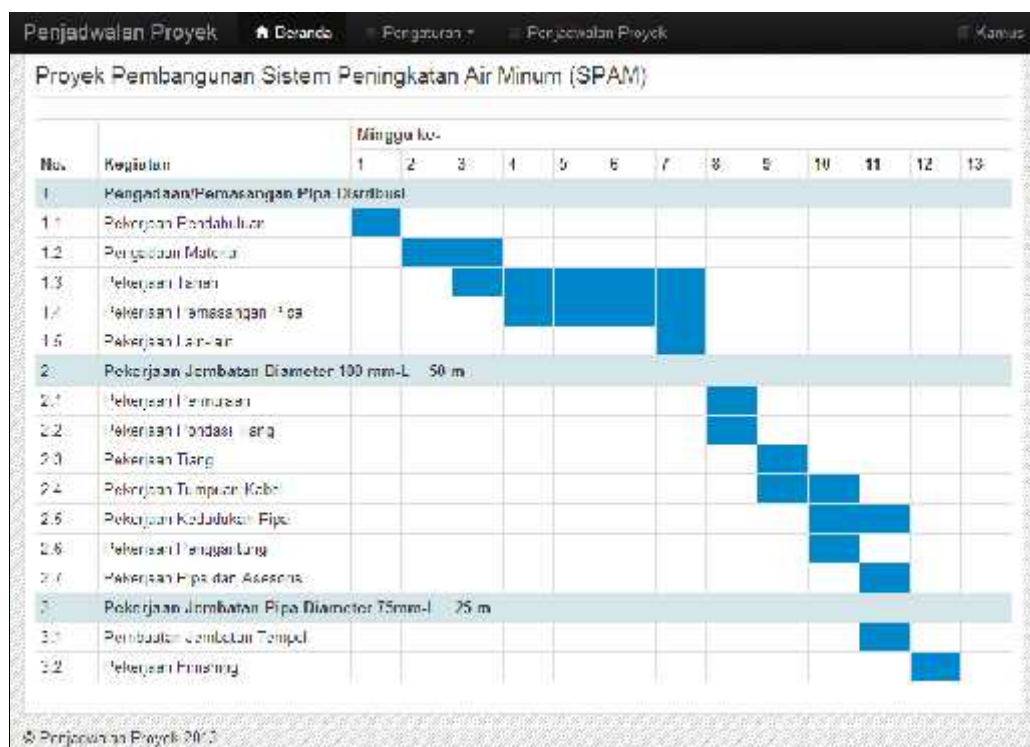
Proyek Pembangunan Sistem Peningkatan Air Minum (SPAM)

No.	Kromosom	Proses	Hasil
1	2	regenerasi ke 7	10
2	1	regenerasi ke 10	10

© Penjadwalan Proyek 2013

Gambar 5.15. Hasil Implementasi *Interface* Kromosom Terbaik

Untuk menampilkan hasil proses penjadwalan proyek menggunakan algoritma genetika dalam bentuk kalender kerja, klik tombol hasil yang terdapat pada tampilan Gambar 5.15 diatas.



Penjadwalan Proyek

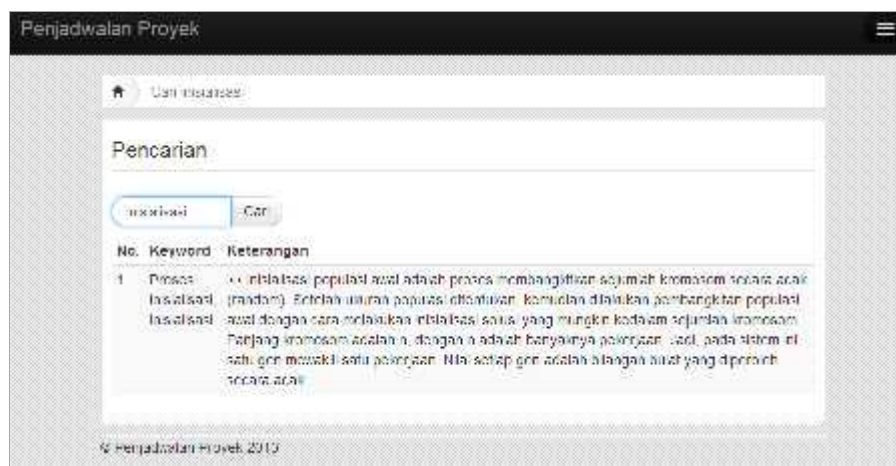
Proyek Pembangunan Sistem Peningkatan Air Minum (SPAM)

No.	Kegiatan	Minggu ke-
1	Pengadaan/Pemasangan Pipa Distribusi	1
1.1	Pekerjaan Pondasi	1
1.2	Pengadaan Material	1
1.3	Pekerjaan Lantai	1
1.4	Pekerjaan Pemasangan Pipa	1
1.5	Pekerjaan Lantai	1
2	Pekerjaan Jambatan Diameter 100 mm-L 50 m	1
2.1	Pekerjaan Pemasangan	1
2.2	Pekerjaan Pondasi	1
2.3	Pekerjaan Tiang	1
2.4	Pekerjaan Tumpuan Kabin	1
2.5	Pekerjaan Kedudukan Pipa	1
2.6	Pekerjaan Pengisian Tangki	1
2.7	Pekerjaan Pipa dan Aksesori	1
3	Pekerjaan Jambatan Pipa Diameter 75mm-L 25 m	1
3.1	Pembuatan Jambatan Tempel	1
3.2	Pekerjaan Finishing	1

© Penjadwalan Proyek 2013

Gambar 5.16. Hasil Implementasi *Interface* Kalender Kerja.

#### 4. Implementasi *Interface* Menu Kamus



Gambar 5.17. Hasil Implementasi *Interface* Menu Kamus

## 5.2. Pengujian Sistem

Setelah tahap implementasi selesai, maka dilanjutkan dengan pengujian dari implementasi yang telah dibuat. Pengujian sistem dilakukan dengan tujuan untuk menjamin sistem yang dibangun sesuai dengan hasil analisa dan perancangan sehingga dapat dibuat satu kesimpulan akhir. Pengujian fungsionalitas sistem dilakukan dengan metode *blackbox*.

### 5.2.1. Pengujian *Blackbox*

Pengujian sistem dilakukan untuk memeriksa kinerja antar komponen sistem yang diimplementasikan. Tujuan utama dari pengujian sistem adalah untuk memastikan bahwa elemen-elemen atau komponen-komponen dari sistem telah berfungsi sesuai dengan yang diharapkan. Salah satu metode pengujian jenis ini dikenal dengan pengujian *blackbox*. Adapun hasil dari pengujian ini dapat dilihat pada tabel 5.1.



Tabel 5.1. Hasil Pengujian Sistem dengan Metode *Blackbox*

No	Objek Pengujian	Hasil yang diharapkan	Hasil
1.	<i>Button</i> Simpan Data pada submenu proyek.	Menyimpan data proyek yang telah diinputkan kedalam database.	Sukses
2.	<i>Button</i> Reset Data pada submenu proyek.	Mengembalikan ke data semula.	Sukses
3.	<i>Button</i> Tambah, Ubah, dan Hapus pada submenu kegiatan.	Tambah : menampilkan halaman tambah kegiatan. Ubah : mengubah data kegiatan. Hapus : menghapus data kegiatan.	Sukses
4.	<i>Button</i> Simpan Data pada halaman Tambah Kegiatan.	Menyimpan data kegiatan yang telah diinputkan kedalam database.	Sukses
5.	<i>Button</i> Reset Data pada halaman Tambah Kegiatan.	Mengembalikan ke data semula.	Sukses
6.	<i>Button</i> Tambah, Ubah, Hapus, dan Hitung Nilai Fitness pada menu Penjadwalan Proyek Proses Inisialisasi.	Tambah : menampilkan halaman tambah kromosom. Ubah : mengubah data kromosom. Hapus : menghapus data kromosom. Hitung Nilai <i>Fitness</i> : menampilkan halaman proses selanjutnya.	Sukses
7.	<i>Button</i> Simpan Data pada halaman Tambah Kromosom.	Menyimpan data kromosom yang telah diinputkan kedalam database.	Sukses

8.	<i>Button</i> Reset Data pada halaman Tambah Kromosom.	Mengembalikan ke data semula.	Sukses
9.	<i>Button</i> Proses Seleksi pada menu Penjadwalan Proyek Proses Evaluasi.	Menampilkan halaman proses selanjutnya.	Sukses
10.	<i>Button Crossover</i> pada menu Penjadwalan Proyek Proses Seleksi.	Menampilkan halaman proses selanjutnya.	Sukses
11.	<i>Button Cancel</i> pada form input nilai probabilitas <i>crossover</i> .	Mengembalikan ke halaman sebelumnya.	Sukses
12.	<i>Button</i> Proses <i>Crossover</i> pada form input nilai probabilitas <i>crossover</i> .	Menampilkan halaman proses pindah silang ( <i>crossover</i> ).	Sukses
13.	<i>Button</i> Mutasi pada menu Penjadwalan Proyek Proses <i>Crossover</i> .	Menampilkan halaman proses selanjutnya.	Sukses
14.	<i>Button</i> Proses Mutasi pada form input nilai probabilitas mutasi.	Menampilkan halaman proses mutasi.	Sukses



15.	<i>Button</i> Re-Generasi pada pada menu Penjadwalan Proyek Proses Mutasi.	Menampilkan form input jumlah generasi.	Sukses
16.	<i>Button</i> Proses Re-Generasi pada form input jumlah generasi.	Memproses sebanyak jumlah generasi yang diinputkan dan menampilkan hasil proses penjadwalan.	Sukses
17.	<i>Button</i> Hasil pada menu Penjadwalan Proyek Proses Generasi.	Menampilkan kalender kerja proyek.	Sukses

Dari pengujian *blackbox* diatas dapat disimpulkan bahwa fitur-fitur yang terdapat pada sistem dapat berjalan dengan baik dan sesuai dengan hasil yang diharapkan.

### 5.2.2. Pengujian Performansi

Pengujian performansi merupakan pengujian yang dilakukan sebanyak 10 kali menggunakan nilai parameter *default* dan nilai parameter masukan dari *user* terhadap sistem penjadwalan proyek. Pengujian ini dilakukan untuk mengetahui apakah aplikasi penjadwalan proyek menggunakan algoritma genetika berhasil menghasilkan jadwal proyek yang memenuhi semua *constraint*. Hasil dari pengujian ini dapat dilihat pada tabel 5.2.

#### 1. Pengujian Performansi Menggunakan Nilai Parameter *Default*

Pengujian ini menggunakan nilai parameter *default*, yaitu:

- Probabilitas *Crossover* ( $P_c$ ) = 0.9
- Probabilitas Mutasi ( $P_m$ ) = 0.1

Berikut hasil pengujian dari proses penjadwalan proyek menggunakan algoritma genetika dengan 10 kali generasi, dapat dijelaskan pada tabel dibawah ini.

Tabel 5.2 Pengujian Performansi Menggunakan Nilai Parameter *Default*

	Kromosom Terbaik	Generasi Ke-	Solusi/ Isi Kromosom	Hasil Pengujian
1	Kromosom 2	7	1 2 3 4 7 8 8 9 9 10 10 11 11 12	Berhasil
2	Kromosom 1	3	1 2 3 4 5 8 8 9 9 10 10 11 11 12	Berhasil
3	Kromosom 4	1	1 2 2 5 8 9 9 10 10 11 11 12 12 13	Berhasil
4	Kromosom 7	5	1 2 3 4 5 8 8 9 9 10 10 11 11 12	Berhasil
5	Kromosom 8	3	1 2 3 4 7 8 8 9 9 10 10 11 11 12	Berhasil
6	Kromosom 3	8	1 2 2 5 8 9 9 10 10 11 11 12 12 13	Berhasil
7	Kromosom 2	6	1 2 3 4 5 8 8 9 9 10 10 11 11 12	Berhasil
8	Kromosom 1	7	1 2 2 5 8 9 9 10 10 11 11 12 12 13	Berhasil
9	Kromosom 6	2	1 2 3 4 5 8 8 9 9 10 10 11 11 12	Berhasil
10	Kromosom 9	1	1 2 3 4 7 8 8 9 9 10 10 11 11 12	Berhasil

Berdasarkan Tabel 5.2 dapat dijelaskan bahwa pengujian pertama menghasilkan kromosom 2 pada generasi ke-7 sebagai solusi penjadwalan proyek yang tidak terdapat pelanggaran terhadap *constraint*. Demikian juga dengan pengujian kedua sampai dengan pengujian kesepuluh.

## 2. Pengujian Performansi Menggunakan Nilai Parameter Inputan *User*

Pengujian ini menggunakan nilai parameter yang diinputkan oleh pengguna (*user*) sistem. Berikut hasil pengujiannya dijelaskan pada tabel dibawah ini.

Tabel 5.3 Pengujian Performansi Menggunakan Nilai Parameter Inputan *User*

	Pc	Pm	Generasi	Kromosom	Solusi/ Isi Kromosom	Waktu yang dibutuhkan	Hasil Pengujian
1	0.25	0.1	20	3	1 2 2 5 8 9 9 10 10 11 11 12 12 13	1 menit 58 detik	Berhasil
2	0.25	0.9	25	1	1 2 3 4 5 8 8 9 9 10 10 11 11 12	2 menit 10 detik	Berhasil
3	0.5	0.8	30	2	1 2 3 4 7 8 8 9 9 10 10 11 11 12	2 menit 22 detik	Berhasil
4	0.5	0.7	40	1	1 2 3 4 5 9 9 10 10 11 11 12 12 13	2 menit 46 detik	Berhasil
5	0.75	0.6	50	6	1 2 2 5 8 9 9 10 10	3 menit 10	Berhasil

					11 11 12 12 13	detik	
6	0.75	0.5	70	5	1 2 3 4 5 8 8 9 9 10 10 11 11 12	3 menit 58 detik	Berhasil
7	0.8	0.4	80	4	1 2 3 4 7 8 8 9 9 10 10 11 11 12	4 menit 22 detik	Berhasil
8	0.8	0.3	90	8	1 2 2 5 8 9 9 10 10 11 11 12 12 13	4 menit 46 detik	Berhasil
9	0.9	0.2	100	7	1 2 3 4 5 8 8 9 9 10 10 11 11 12	5 menit 56 detik	Berhasil
10	0.9	0.1	120	3	1 2 3 4 7 8 8 9 9 10 10 11 11 12	7 menit 40 detik	Berhasil

Berdasarkan Tabel 5.3 dapat dijelaskan bahwa pengujian pertama dengan nilai probabilitas *crossover* ( $P_c$ ) = 0.25 dan probabilitas mutasi ( $P_m$ ) = 0.1 sebanyak 20 generasi menghasilkan kromosom 3 sebagai solusi penjadwalan proyek yang memenuhi semua *constraint* dengan membutuhkan waktu 1 menit 58 detik untuk penggenerasiannya. Demikian juga dengan pengujian kedua sampai dengan pengujian kesepuluh.

### 5.2.3. User Acceptance Test

*User Acceptance Test* merupakan pengujian yang dilakukan dengan meminta persetujuan dari *user* terhadap *output* yang dihasilkan oleh sistem penjadwalan proyek. Responden yang melakukan pengujian yaitu sebanyak 10 orang. Pengujian *User Acceptance Test* selengkapnya dapat dilihat pada lampiran B. Berikut hasil pengujiannya dijelaskan pada Tabel 5.4 dibawah ini.

Tabel 5.4 Hasil pengujian *User Acceptance Test*

No	Butir Pengujian	Hasil Pengujian		Responden yang setuju
		Ya	Tidak	
1	Penggunaan sistem penjadwalan proyek sebelumnya.			10 Orang
2	Pernah melihat Sistem Penjadwalan Proyek menggunakan Algoritma Genetika			10 Orang
3	Tampilan ( <i>interface</i> )			10 Orang

	Sistem membuat bosan			
4	Menu-menu dalam sistem sulit digunakan			10 Orang
5	Sistem dapat membantu pencarian solusi jadwal proyek			10 Orang
6	Warna dan gambar yang digunakan sistem cocok dan serasi			10 Orang
7	Sistem dapat memberikan solusi penjadwalan proyek			10 Orang
8	Terdapat <i>error</i> pada sistem			10 Orang
9	Hasil dari sistem memuaskan			10 Orang
10	Sistem ini tetap digunakan untuk jangka waktu yang akan datang			10 Orang

Berdasarkan Tabel 5.4 dapat dijelaskan bahwa 10 orang responden menyetujui sistem penjadwalan proyek menggunakan algoritma genetika dapat berjalan dengan baik sesuai dengan yang diharapkan dan tidak sulit dalam penggunaannya.

#### **5.2.4. Hasil Pengujian**

Dari pengujian sistem yang telah dilakukan, sistem penjadwalan proyek menggunakan algoritma genetika dapat menghasilkan jadwal proyek yang optimal karena tidak terdapat pelanggaran terhadap aturan penjadwalan proyek yang telah ditetapkan.

#### **5.2.5. Kesimpulan Pengujian**

Berdasarkan pengujian yang telah dilakukan sebanyak 10 kali pengujian dapat diambil beberapa kesimpulan, yaitu sebagai berikut :

- a. Pengujian *Balckbox* pada Tabel 5.1 menunjukkan bahwa komponen-komponen dari sistem telah berfungsi sesuai dengan yang diharapkan.
- b. Pengujian performansi yang dilakukan dengan menggunakan nilai parameter *default* pada Tabel 5.2 dan inputan dari *user* pada Tabel 5.3 berhasil berjalan dengan baik.
- c. Berdasarkan pengujian performansi pada Tabel 5.3 juga dapat disimpulkan bahwa algoritma genetika membutuhkan waktu yang lama jika nilai iterasinya besar, karena dalam prosesnya algoritma ini melakukan proses penggenerasian.
- d. Pengujian dengan *User Acceptance Test* pada Tabel 5.4 yang dilakukan kepada 10 orang responden yang telah menggunakan Sistem Penjadwalan Proyek menggunakan Algoritma Genetika menunjukkan bahwa *output* yang dihasilkan sistem dapat menjadi solusi dalam menjadwalkan pekerjaan atau kegiatan dalam sebuah proyek.

## **BAB VI**

### **PENUTUP**

#### **6.1. Kesimpulan**

Setelah menyelesaikan serangkaian tahapan-tahapan dalam menerapkan algoritma genetika pada sistem penjadwalan proyek maka dapat ditarik kesimpulan yaitu:

1. Sistem penjadwalan proyek menggunakan algoritma genetika ini telah berhasil dibangun dan diimplementasikan untuk mencari solusi jadwal yang optimal yaitu memenuhi batasan/persyaratan (*constraints*) dalam penjadwalan proyek, meliputi beberapa kegiatan tidak bisa dimulai sebelum kegiatan yang lain selesai, mengoptimalkan efisiensi pemakaian waktu pelaksanaan setiap kegiatan, dan tidak boleh melebihi durasi proyek yang telah ditetapkan.
2. Sistem penjadwalan proyek ini mampu menangani operasi *input* data, melakukan proses penjadwalan proyek secara otomatis dan menghasilkan jadwal yang optimal serta mencetak kalender kerja.
3. Algoritma genetika bekerja berdasarkan suatu bilangan acak (*random*) yang dibangkitkan pada masing-masing solusi sehingga menyebabkan hasil analisa akan berbeda dengan hasil pengujian menggunakan sistem yang dibangun.
4. Dari hasil *user acceptance test* dapat disimpulkan bahwa sistem penjadwalan proyek menggunakan algoritma genetika dapat memberikan solusi penjadwalan kegiatan-kegiatan dalam sebuah proyek yang dapat menjadi alternatif keputusan bagi kontraktor dalam pelaksanaan proyek.

#### **6.2. Saran**

Adapun saran-saran yang diajukan oleh penulis untuk pengembangan penelitian selanjutnya yaitu:

1. Dapat dikembangkan sistem penjadwalan proyek yang dapat menangani berbagai kendala, seperti keterbatasan sumber daya manusia dan biaya.
2. Diharapkan sistem penjadwalan proyek hendaknya dapat memberikan lebih banyak pilihan metode seleksi dan metode pindah silang (*crossover*).

## DAFTAR PUSTAKA

- Abdul, Kadir. *Pengenalan Sistem Informasi*. Yogyakarta, Andi, 2003.
- Afandi, Fachrudin. “Penerapan Algoritma Genetika Untuk Masalah Penjadwalan *Job Shop* pada Lingkungan Industri Pakaian”. Skripsi Sarjana, Fakultas Teknologi Informasi Institut Teknologi Sepuluh November (ITS), Surabaya, 2009.
- Arifudin, Riza. “Optimasi Penjadwalan Proyek dengan Penyeimbangan Biaya Menggunakan Kombinasi CPM dan Algoritma Genetika”. Jurnal Masyarakat Informatika, Volume 2, Nomor 4, ISSN 2086 – 4930.
- D. Naso, M. Surico, B.Turchiano, dan U. Kaymak. “ Genetic algorithms for supply-chain scheduling : A case study in the distribution of ready-mixed concrete”. *European Journal of Operational Research*, 2006.
- Erika, Maya. “Sistem Perencanaan dan Pengendalian Jadwal Proyek Menggunakan Precedence Diagram Method (PDM)”. Skripsi Sarjana, Fakultas Sains dan Teknologi Universitas Islam Negeri SUSKA, Riau, 2008.
- Fadlisyah, Arnawan, Faisal. *Algoritma Genetika*. Graha Ilmu, Yogyakarta, 2009.
- Hartono, Jogiyanto. *Analisis & Desain Sistem Informasi: Pendekatan terstruktur teori dan praktek aplikasi bisnis*. Andi Offset, Yogyakarta, 1990.
- Hartono, Jogiyanto. *Analisis & Desain Sistem Informasi: Pendekatan terstruktur teori dan praktek aplikasi bisnis*. Andi Offset, Yogyakarta, 2005.
- Kusumadewi, Sri. *Articial Intelligence*. Graha Ilmu, Yogyakarta, 2003.
- Putra, Yendrika. “Aplikasi Penjadwalan Perkuliahan Menggunakan Algoritma Genetika”. Skripsi Sarjana, Fakultas Sains dan Teknologi Universitas Islam Negeri SUSKA, Riau, 2009.
- Sam’ani. “Rancang Bangun Sistem Penjadwalan Perkuliahan dan Ujian Akhir Semester dengan Pendekatan Algoritma Genetika”. Tesis Pascasarjana, Universitas Diponegoro, Semarang, 2012.
- Suyanto. *Algoritma Genetika dalam Matlab*. Andi, Yogyakarta, 2005.